

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

SISTEMA INFORMÁTICO APLICANDO PYTHON PARA LA GESTIÓN DE PRODUCTOS CON SERIES DE TIEMPO EN LA PANADERÍA Y PASTELERÍA “FLOR DE CEBADA"

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información

ÁREA: Diseño de sistemas de información

LÍNEA DE INVESTIGACIÓN: Tecnologías de la investigación y Sistemas de control

AUTOR: Jean Carlo Freire Valencia

TUTOR: Ing. Clay Fernando Aldas Flores, Mg

Ambato - Ecuador

febrero - 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA INFORMÁTICO APLICANDO PYTHON PARA LA GESTIÓN DE PRODUCTOS CON SERIES DE TIEMPO EN LA PANADERÍA Y PASTELERÍA “FLOR DE CEBADA" desarrollado bajo la modalidad Proyecto de Investigación por el señor Jean Carlo Freire Valencia, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.

Firmado electrónicamente por:
CLAY FERNANDO ALDAS FLORES

Ing. Clay Fernando Aldas Flores, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA INFORMÁTICO APLICANDO PYTHON PARA LA GESTIÓN DE PRODUCTOS CON SERIES DE TIEMPO EN LA PANADERÍA Y PASTELERÍA “FLOR DE CEBADA" es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.



Jean Carlo Freire Valencia

C.C. 1805102066

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la Institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.



Jean Carlo Freire Valencia

C.C. 1805102066

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jean Carlo Freire Valencia, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA INFORMÁTICO APLICANDO PYTHON PARA LA GESTIÓN DE PRODUCTOS CON SERIES DE TIEMPO EN LA PANADERÍA Y PASTELERÍA “FLOR DE CEBADA", nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.

Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL

SANTIAGO DAVID JARA MOYA

Firmado digitalmente por SANTIAGO DAVID JARA MOYA
Fecha: 2025.02.03 08:23:06 -05'00'

Ing. Santiago David Jara Moya Mg.

PROFESOR CALIFICADOR

Firmado electrónicamente por:
LEONARDO DAVID TORRES VALVERDE

Ing. Leonardo David Torres Valverde Mg.

PROFESOR CALIFICADOR

DEDICATORIA

A mi familia, por su amor incondicional, apoyo constante y por enseñarme que el esfuerzo y la perseverancia son la clave para alcanzar cualquier meta. Gracias por estar siempre a mi lado, brindándome su comprensión y motivación.

A mis maestros, por su dedicación y paciencia. Por compartir sus conocimientos y enseñanzas, y por inspirarme a seguir aprendiendo y creciendo en cada paso. Su influencia ha sido fundamental en mi formación académica y personal.

Jean Carlo Freire Valencia

AGRADECIMIENTO

En primer lugar, a Dios y mi familia, por su apoyo incondicional, amor y comprensión. Gracias por ser mi fuente de fortaleza y por siempre creer en mí, incluso cuando el camino se volvía incierto. Su paciencia y sacrificio me han permitido llegar hasta aquí.

A mis maestros y mi tutor, por su guía y dedicación. Sus enseñanzas no solo me han proporcionado conocimientos, sino también una visión más amplia y profunda sobre mi campo de estudio. Gracias por su tiempo, paciencia y por siempre alentarme a superarme.

A mis amigos, por su apoyo emocional y por compartir este viaje académico. Gracias por las conversaciones, los momentos de colaboración y por ser una fuente de inspiración.

Jean Carlo Freire Valencia

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR ..... **ii**
DEDICATORIA ..... **vi**
AGRADECIMIENTO ..... **vii**
ÍNDICE GENERAL DE CONTENIDOS ..... **viii**
ÍNDICE DE FIGURAS ..... **x**
ÍNDICE DE TABLAS ..... **xii**
RESUMEN EJECUTIVO ..... **xiv**
ABSTRACT ..... **xv**
CAPÍTULO I. MARCO TEÓRICO **16**
1.1 Tema de investigación 16
1.1.1 Planteamiento del problema 16
1.2 Antecedentes investigativos 17
1.3 Fundamentación teórica 21
1.4 Objetivos 27
1.4.1 Objetivo general 27
1.4.2 Objetivos específicos 27
CAPÍTULO II. METODOLOGÍA **28**
2.1 Materiales 28
2.2 Métodos 32
2.2.1 Modalidad de la investigación 32
2.2.2 Población y muestra 32
2.2.3 Recolección de información 33
2.2.4 Procesamiento y análisis de datos 45
CAPÍTULO III. RESULTADOS Y DISCUSIÓN **47**
3.1 Análisis y discusión de los resultados 47
3.1.1 Análisis de gestión de procesos 47
3.1.2 Aplicación de machine learning en el uso de series de tiempo 51
3.1.4 Definición de la metodología 59
3.2 Desarrollo de la propuesta 61
3.2.1 Fase I: Planificación 61
3.2.2. Fase II: Diseño 69

3.2.3. Fase III: Codificación 86
3.2.4. Fase IV: Pruebas 105
3.2.5. Fase V: Implantación 113
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 119
4.1 Conclusiones 119
4.2 Recomendaciones 120
REFERENCIAS BIBLIOGRÁFICAS 121
ANEXOS 124
Anexo A manual de Usuario 124

ÍNDICE DE FIGURAS

Figura 1. Frecuencia con la que el trabajador revisa la disponibilidad de los productos. 35
Figura 2. Forma actual de manejo de los productos de la panadería 36
Figura 3. Problemas que encuentra el trabajador con el manejo de los productos. .. 36
Figura 4. La automatización del sistema ayudaría al trabajo. 37
Figura 5. Ha usado una herramienta de gestión de productos. 38
Figura 6. Comodidad del trabajado al usar una herramienta de gestión de productos. 38
Figura 7. Opciones que le gustaría al trabajador para la gestión de productos. 39
Figura 8. Frecuencia con la que el trabajador tiene problemas con productos. 40
Figura 9. lo que sucede cuando un producto caduco en la panadería. 40
Figura 10. Importancia de los productos de los datos en el futuro. 41
Figura 11. Análisis de procesos actual. 47
Figura 12. análisis de procesos actualizado. 49
Figura 13. Capas de arquitectura de la aplicación. 63
Figura 14. Diagrama de base de datos. 79
Figura 15. Control de acceso al sistema. 80
Figura 16. Registro de nuevos usuarios. 80
Figura 17. Diseño del dashboard principal. 81
Figura 18. Visualización de productos con bajo stock y próximos a caducar. 82
Figura 19. Gestión de inventario. 82
Figura 20. Gestión de ventas. 83
Figura 21. Generación de nota de venta en PDF. 83
Figura 22. Generación de reportes. 84
Figura 23. Predicciones de demanda. 85
Figura 24**.** Visualización del historial de ventas. 85
Figura 25. Creación de solución backend. 86
Figura 26. Creación de solución en frontend. 87
Figura 27. Creación de Base de datos. 87
Figura 28. Librerías usadas en backend. 88
Figura 29. Método registrar usuarios. 88
Figura 30. Método login. 89
Figura 31. Método registrar usuario. 89
Figura 32. Método obtener productos. 90
Figura 33. Método añadir un nuevo producto. 90
Figura 34. Método actualizar productos. 91
Figura 35 .Método eliminar producto. 91
Figura 36. Método buscar producto por código de barras. 92
Figura 37. Método generar PDF. 92
Figura 38. Método actualizar stock después de la venta. 93
Figura 39. Método de registrar una venta. 94
Figura 40. Método para obtener detalle de ventas realizadas. 94

Figura 41. Método para buscar productos por nombre o código de barras. 95
Figura 42. Método para verificar bajo stock. 96
Figura 43. Método para obtener productor cerca de caducar. 96
Figura 44. Método para crear grafico de tendencias. 97
Figura 45. Método de predicción de la demanda. 98
Figura 46. Método para generación de gráficos. 98
Figura 47. Método de limpieza de gráficos. 99
Figura 48. Método de generación de reportes en PDF. 99
Figura 49. Componentes del frontend. 100
Figura 50. Clase Dashboard.js. 101
Figura 51. Clase GeneracionReportes.js. 101
Figura 52. Clase GestionProductos.js. 102
Figura 53. Clase Inicio.js. 102
Figura 54. Clase Login.js. 103
Figura 55. Clase PrivateRoute.js. 103
Figura 56. Clase RegistroU.js. 104
Figura 57. Clase ReportedeVentas.js. 104
Figura 58. Clase de Ventas.js. 105
Figura 59. Creación del build para el backend. 113
Figura 60. Instalación y configuración del servidor. 113
Figura 61. Script ini.bat. 114
Figura 62. Script inifront.bat. 114
Figura 63. Script inibak.bat. 114
Figura 64. Script iniweb.bat. 115
Figura 65. Configuración general de la tarea programada. 115
Figura 66. Acciones con la ruta para el inicio de la aplicación. 116
Figura 67. Desencadenador de la tarea programada. 116
Figura 68. Inicio del sistema. 117

ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda aplicadas en base de datos científicas. 18
Tabla 2. Filtro de búsqueda aplicado en repositorio de universidades de Ecuador. .. 18
Tabla 3. Criterios de inclusión y exclusión. 19
Tabla 4. Modelo de Ficha de observación. 32
Tabla 5. Población de estudio. 33
Tabla 6. Cálculo y sumatoria de varianza de la encuesta. 34
Tabla 7. Cálculo de Alfa de Cronbach. 34
Tabla 8. Tabla de confiabilidad de Alfa de Cronbach. 34
Tabla 9. Resultado de la ficha de observación. 44
Tabla 10. Análisis del paper **"**Inventory Management and Cost Reduction of Supply Chain Processes Using A-Based Time-Series Forecasting and ANN Modeling"[20]. 53
Tabla 11. Análisis del paper **"**Optimization of inventory management through computer vision and machine learning technologies"[21]. 53
Tabla 12. Análisis del paper **"**Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis"[22]. 54
Tabla 13. Comparación de modelos utilizados en los distintos casos de estudio. 55
Tabla 14. Tabla comparativa de bases de datos. 56
Tabla 15. Cuadro comparativo de tecnologías frontend. 59
Tabla 16. Tabla comparativa de metodologías agiles. 60
Tabla 17. Cuadro de definición de roles. 63
Tabla 18. Modelo de historias de usuario. 65
Tabla 19. Historia de Control de acceso del sistema. 65
Tabla 20. Historia Registro de nuevos Usuarios. 66
Tabla 21. Historia de diseño de dashboard principal. 66
Tabla 22. Historia de Control de productos de caducidad y de stock. 66
Tabla 23. Historia de gestión de inventario. 67
Tabla 24. Historia de gestión de ventas. 67
Tabla 25. Historia de generación de nota en pdf. 67
Tabla 26. Historia de generación de reportes. 67
Tabla 27. Historia de control de acceso del sistema. 68
Tabla 28. Historia de visualización del historial de ventas. 68
Tabla 29. Valoración de historias de usuario. 69
Tabla 30. Plan de entrega. 69
Tabla 31. Historia de control de iteración 1. 70
Tabla 32. Actividades de ingreso del sistema. 70
Tabla 33. Actividades de registro de nuevos usuarios. 70
Tabla 34. Actividades de diseño del dashboard principal. 71
Tabla 35. Historia de control de iteración 2. 71
Tabla 36. Actividades de visualización de productos con bajo stock y próximos a caducar. 71

Tabla 37. Actividades de gestión de inventario. 72
Tabla 38. Actividades de gestión de ventas. 72
Tabla 39. Actividades de generación de nota de venta en PDF. 72
Tabla 40. Historia de control de iteración 3. 73
Tabla 41. Actividades de generación de reportes. 73
Tabla 42. Actividades de predicción de la demanda. 73
Tabla 43. Visualización del historial de ventas. 74
Tabla 44. Tarjeta CRC de ingreso al sistema. 74
Tabla 45. Tarjeta CRC de registro de nuevos usuarios. 75
Tabla 46. Tarjeta CRC de dashboard principal. 75
Tabla 47. Tarjeta CRC de visualización de productos con bajo stock y próximos a caducar. 75
Tabla 48. Tarjeta CRC de gestión de inventario. 76
Tabla 49. Tarjeta CRC de gestión de ventas. 76
Tabla 50. Tarjeta CRC de generación de nota de venta PDF. 77
Tabla 51. Tarjeta CRC de generación de reportes. 77
Tabla 52. Tarjeta CRC de predicción de la demanda. 78
Tabla 53. Tarjeta CRC de historial de ventas. 78
Tabla 54. Prueba de aceptación de control de acceso al sistema. 106
Tabla 55. Prueba de aceptación de registro de nuevos usuarios. 106
Tabla 56. Prueba de aceptación de diseño del dashboard principal. 107
Tabla 57. Prueba de aceptación de visualización de productos con bajo stock y próximos a caducar. 107
Tabla 58. Prueba de aceptación de gestión de inventario. 108
Tabla 59. Prueba de aceptación de gestión de ventas. 108
Tabla 60. Prueba de aceptación de generación de nota de venta en PDF. 109
Tabla 61. Prueba de aceptación de generación de reportes. 109
Tabla 62. Prueba de aceptación de Predicciones de demanda. 110
Tabla 63. Prueba de aceptación de visualización del historial de ventas. 110
Tabla 64. Resultado de pruebas de usabilidad. 112
Tabla 65. Capacitación a la panadería y pastelería “Flor de Cebada”. 118

RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo desarrollar un sistema de software para mejorar la gestión de productos en la panadería “Flor de Cebada”. Utilizando Python y el framework flask para el backend, Machine Learning y React para el frontend, el sistema integra la previsión de series temporales para optimizar la detección de tendencias y la predicción de la demanda. Esto ayuda a mejorar las ventas y tomar decisiones basadas en datos, en un entorno de pequeña empresa.

El sistema gestiona el inventario mediante el registro, actualización y seguimiento de productos, incluyendo detalles como cantidades, fechas de vencimiento y precios. Además, agiliza los procesos de venta con escaneo de códigos de barras y cálculos automáticos. Genera informes y proporciona previsiones precisas de la demanda para alinear la producción con las necesidades del mercado. También emite recibos electrónicos para cada transacción, mejorando la eficiencia operativa.

El desarrollo del proyecto toma como fundamento la metodología Extreme Programming, destacando el diseño iterativo, la colaboración y la adaptabilidad. El sistema beneficia las operaciones comerciales, mejora la planificación del inventario, aumenta la satisfacción del cliente y establece una base para el crecimiento sostenible de la panadería, demostrando el valor de las soluciones tecnológicas en pequeñas y medianas empresas.

Palabras Clave: Python, Flask, Machine learning, Predicción de la demanda, Gestión de inventarios.

ABSTRACT

The present project focuses on the development of a software system aimed at improving product management processes in the bakery and pastry shop "Flor de Cebada." Leveraging Python, the Flask framework for the backend, and React for the frontend, alongside the application of machine learning, the system integrates time series forecasting to enhance trend detection and demand prediction. The goal is to optimize sales operations, minimize product waste, and support data-driven decision-making in a small business environment.

The system includes an inventory management mechanism, enabling the registration, updating, and tracking of products, including essential details such as quantities, expiration dates, and prices. Additionally, sales processes are streamlined through barcode scanning, automatic calculations, and totals.

A key feature is the generation of reports required by the business. The predictive analysis implemented in the system development provides accurate demand forecasts, helping align production with market needs. The system also generates electronic receipts for transactions, improving operational efficiency and user experience.

The project was developed using the Extreme Programming methodology, emphasizing iterative design, stakeholder collaboration, and adaptability to changing requirements. The system delivers tangible benefits for business operations, enabling the bakery to achieve greater accuracy in inventory planning, enhance customer satisfaction, and establish a foundation for sustainable growth. This initiative highlights the potential of technology-driven solutions to address real-world challenges faced by small and medium-sized enterprises.

Keywords: Python, Flask, Machine learning, Demand forecasting, Inventory management.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA INFORMÁTICO APLICANDO PYTHON PARA LA GESTIÓN DE PRODUCTOS CON SERIES DE TIEMPO EN LA PANADERÍA Y PASTELERÍA “FLOR DE CEBADA".

1.1.1 Planteamiento del problema

En el contexto global, preocupa que el sistema lineal de producción en el sector alimentario genere grandes cantidades de desperdicio. Se estima que a nivel mundial se desperdicia el equivalente a seis camiones de comida por segundo, con niveles per cápita de desperdicio en Europa y Norteamérica entre 95-115 kg/año, más de diez veces los niveles de África subsahariana y el sur/sudeste de Asia[1]. Este problema se ve agravado por el hecho de que se espera que para el año 2050, el 80% de todos los alimentos se consuman en zonas urbanas, lo que podría empeorar la situación si no se toman medidas urgentes.

En el caso de la gestión de productos alimentarios, se pueden utilizar técnicas de Inteligencia Artificial (IA) para predecir la demanda de alimentos en diferentes procesos productivos, utilizando modelos de aprendizaje automático clásicos y modelos de deep learning basados en datos históricos, como producción y variables ambientales [2].

En Latinoamérica, la gestión de productos alimentarios se beneficia del uso de la IA y las series de tiempo para predecir la demanda, considerando factores como el clima y las festividades. Los modelos de aprendizaje automático y deep learning se utilizan para analizar datos históricos y variables ambientales, permitiendo predecir la demanda de alimentos en diferentes procesos productivos.

El sector de la panadería y pastelería es una industria en crecimiento a nivel latinoamericano, con un crecimiento del 4.1%[3]. Según la Organización Mundial de la Salud, se consume un promedio de 37 kilos de pan por persona anualmente, sin

embargo, en Ecuador, según un informe de Levapan, solo se consumen 28 kilos por persona[3]. Esta disparidad en el consumo puede atribuirse a diversas razones, como preferencias alimentarias locales y la disponibilidad de productos en el mercado.

En el caso específico de Ecuador, la producción nacional de trigo solo cubre el 2% del consumo local, lo que significa que el 98% es importado[4]. Esto destaca la importancia de asegurar un suministro estable de trigo para la producción de pan y otros productos básicos. En este sentido, el Pleno del Comercio Exterior ha decidido mantener la exención del arancel sobre el trigo, medida que ha estado vigente desde el 2020 y se extenderá hasta el 2024.

La industria panadera y pastelera no solo representa un importante sector económico, sino que también tiene un impacto significativo en la generación de empleo y en la matriz productiva del país. Según el Censo del 2010, existen en Ecuador 8,214 establecimientos dedicados a la panadería y pastelería, generando 22,000 plazas de trabajo y aportando 757.4 millones de dólares a la economía [5].

La panadería y pastelería “Flor de Cebada” con una trayectoria de 4 años en el mercado, es consciente de la importancia de implementar sistemas para mejorar la eficiencia de su establecimiento y reducir sus tiempos de producción.

1.2 Antecedentes investigativos

Se realizó una revisión exhaustiva de la literatura mediante la consulta de diversas fuentes de información científica y académica. Las cadenas de búsqueda se diseñaron cuidadosamente utilizando palabras clave pertinentes, “time series”, “python”, “collection system”, “product management”, en una de las bases de datos más reconocidas, IEEE Xplore. Estas búsquedas se llevaron a cabo entre los años 2019 y 2024, seleccionando exclusivamente los artículos de acceso libre.

Obteniendo los siguientes resultados.

Tabla 1. Cadenas de búsqueda aplicadas en base de datos científicas.

Base de datos	**Cadenas de Búsqueda**	**Núm. de Artículos**
IEEE Xplore	(All Metadata": time series) AND ("All Metadata": product management) Filters Applied: 2019 – 2024 Open Access Only	51
	("All Metadata": Python) AND ("All Metadata": collection system) Filters Applied: 2018 - 2023 Open Access Only	16
	("All Metadata": time series) AND ("All Metadata": Python) Filters Applied: 2018 - 2023 Open Access Only	19
	Total	86

Además, Se realiza una búsqueda en los repositorios de 2 Universidades como son la "Escuela Politécnica Nacional" y la "Escuela Superior Politécnica de Chimborazo" aplicando filtros por carreras y años de publicación ente el 2017 al 2024.

obteniendo los siguientes resultados comprendidos en la **Tabla 2**.

Tabla 2. Filtro de búsqueda aplicado en repositorio de universidades de Ecuador.

Universidad	**Filtro de Búsqueda**	**Núm. de Artículos**
Escuela Superior Politécnica de Chimborazo	Facultad de Informática y Electrónica	529
	Facultad de Administración de Empresas	891
Escuela Politécnica Nacional	Facultad de Ingeniería en Sistemas Informáticos y de Computación	385
	Total	1805

Esta revisión bibliográfica analiza los antecedentes relevantes sobre tecnología y prácticas en el desarrollo de series de tiempo en distinto ámbitos además de analizar la industria de la panadería y pastelería. Se utilizaron criterios específicos para

seleccionar cinco artículos fundamentales que enriquecen a esta investigación, tomando en cuenta lo presentado en la **Tabla 3**.

Tabla 3. Criterios de inclusión y exclusión.

Inclusión	• Artículos que tratan sobre el uso de series de tiempo. • Investigaciones que se centran en desarrollo de software de cobro aplicando metodologías específicas. • Investigaciones sobre diseño organizacional de panaderías en Ecuador.
Exclusión	• Estudios que no tengan que ver con eficiencia operativa. • Artículos que solicitan remuneración económica. • Artículos que no estén relacionados con el diseño organizacional de los negocios.

En su proyecto de investigación, Granizo Peña [6] propuso el diseño organizacional para la panadería Grani's de la ciudad de Riobamba, provincia de Chimborazo. Se basó en la identificación de la división y distribución jerárquica mediante un diagnóstico de la estructura organizacional y el uso de técnicas como entrevistas, encuestas y observaciones, aplicadas al personal de los distintos niveles jerárquicos de la empresa. Los instrumentos utilizados incluyeron guías de entrevistas y cuestionarios con preguntas abiertas y cerradas. La información recolectada proporcionó una idea del funcionamiento de la empresa y permitió levantar el contexto organizacional, identificando fortalezas, debilidades, amenazas y oportunidades. Como resultado, se propuso el diseño organizacional de la panadería Grani's, que incluye organigramas estructurales, funcionales y posicionales, seguidos de un manual de funciones y clasificación de personal.

Ruiz y Paguay [7] propusieron el desarrollo de un sistema web destinado a optimizar el proceso de cobro de agua potable en la comunidad El Socorro. Se basaron en el framework ASP.NET MVC (Model-View-Controller) para separar los datos de la interfaz de usuario y la lógica de negocio, lo que facilitó el mantenimiento y la escalabilidad del sistema. Además, utilizaron la metodología ágil SCRUM para el desarrollo del proyecto. La implementación de la base de datos se realizó en SQL Server 2012. "Mediante el uso de la aplicación web, se determinó que el tiempo en el cálculo del valor a cancelar es de 0,76 segundos, lo que indica una disminución

significativa del tiempo empleado para esta tarea. Además, la aplicación móvil logró una disminución de errores del 92.31%. Se concluye que la utilización del sistema disminuyó el número de errores y el tiempo en un 95.67% en comparación con el método tradicional, por lo que se recomienda su utilización".

En su proyecto de investigación, Ibujés Gómez y Jaramillo Gutierrez [8] propusieron el desarrollo de una aplicación web y móvil para gestionar servicios para mascotas dentro de la ciudad de Quito, denominada Appet. Se basaron en la metodología Scrum, que les permitió guiar el proceso de desarrollo del sistema, desde la recolección de requisitos hasta la implementación, pruebas y aceptación por parte del cliente. El proyecto se desarrolló con la finalidad de brindar una solución a la gestión de servicios tanto para el cliente como para el proveedor, tales como hospedaje, veterinaria, peluquería y tienda para mascotas en los diferentes sectores de la ciudad de Quito. Appet se desarrolló con la herramienta Ionic, un framework de código abierto que permite crear aplicaciones móviles multiplataforma utilizando tecnologías web, y con la plataforma Firebase de Google, que aprovecha el Realtime Database con un modelo de datos intuitivo y de alto rendimiento.

En su proyecto de investigación, Li, Xiong, Zhu, Zhang y Wang [9]. propusieron un método de diagnóstico de fallas basado en la codificación de series temporales y una Convolutional Neural Network (CNN). Se basaron en la metodología Extreme Programming (XP), donde realizaron un levantamiento de información a través de entrevistas y observaciones para comprender los procesos actuales de diagnóstico de fallas en unidades petroquímicas. Los resultados de este estudio revelaron que la implementación de dicho método ha tenido un impacto positivo en la optimización de los procesos de diagnóstico de fallas, mejorando la precisión y la estabilidad en comparación con otras técnicas avanzadas. En conclusión, el método propuesto ha demostrado ser una solución efectiva para mejorar la precisión en el diagnóstico de fallas en unidades petroquímicas.

Siebert, GroB y Schroth [10] realizaron una revisión sistemática de los paquetes de Python con un enfoque en el análisis de series temporales. El objetivo fue proporcionar una visión general de las diferentes tareas de análisis de series temporales y métodos de preprocesamiento implementados, y una visión general de las características de

desarrollo de los paquetes (por ejemplo, documentación, dependencias y tamaño de la comunidad). Esta revisión se basó en una búsqueda en bases de datos de literatura, así como en repositorios de GitHub, la investigación proporciona una guía útil para profesionales e investigadores interesados en el análisis de series temporales en Python, destacando la prevalencia de la predicción como tarea principal.

1.3 Fundamentación teórica

Ingeniería en software

La Ingeniería de Software es una disciplina clave en el desarrollo de sistemas y aplicaciones informáticas, aplicando principios de ingeniería para crear software de alta calidad. En su núcleo, busca garantizar que el software sea confiable, eficiente y cumpla con los requisitos y expectativas del cliente. Este proceso implica una serie de etapas, que van desde la concepción y análisis de requisitos hasta el diseño, implementación, pruebas y mantenimiento del software. Cada etapa requiere un enfoque metódico y sistemático, utilizando herramientas y técnicas especializadas para garantizar que el producto final cumpla con los estándares de calidad y funcionalidad esperados.

La Ingeniería de Software también aborda aspectos como la gestión de proyectos, la calidad del software, la seguridad informática y la ética profesional. Además, se basa en principios como el modularidad, la reutilización de código, la documentación y la colaboración en equipo. Estos principios y prácticas ayudan a garantizar que el software sea desarrollado de manera eficiente, reduciendo costos y tiempos de desarrollo, y aumentando la satisfacción del cliente. En un mundo cada vez más dependiente de la tecnología, la Ingeniería de Software juega un papel fundamental en la creación de sistemas y aplicaciones que impulsen el progreso y mejoren la calidad de vida de las personas [11].

Metodología de desarrollo

Las metodologías de desarrollo de software son enfoques sistemáticos para gestionar el proceso de desarrollo de software, desde la concepción de la idea hasta la entrega y mantenimiento del producto final. Estas metodologías ofrecen un marco de trabajo que

define roles, actividades, artefactos y reglas que guían el desarrollo del software. Una de las metodologías más conocidas es la metodología ágil, que se centra en la colaboración cercana entre equipos multidisciplinarios, la entrega incremental de software funcional y la capacidad de respuesta a cambios en los requisitos del cliente. Otra metodología común es el modelo en cascada, que organiza el desarrollo en etapas secuenciales, como análisis de requisitos, diseño, implementación, pruebas y mantenimiento, con cada etapa dependiendo de la anterior[12].

- **Scrum:** Es una metodología ágil que se centra en la entrega continua de software funcional en iteraciones cortas llamadas sprints. Durante cada sprint, un equipo multidisciplinario trabaja en la planificación, desarrollo, pruebas y entrega de un incremento de software. Scrum se basa en la transparencia, la inspección y la adaptación, lo que permite a los equipos responder rápidamente a los cambios en los requisitos del cliente.

- **Kanban:** Es una metodología de gestión visual que se centra en visualizar el flujo de trabajo y limitar el trabajo en curso para mejorar la eficiencia. Kanban utiliza tableros Kanban para representar visualmente las tareas y su estado en el proceso de desarrollo. Los equipos utilizan tarjetas para representar las tareas y mueven estas tarjetas a través de columnas en el tablero a medida que avanzan en el trabajo.

- **Extreme Programming (XP)**: XP es una metodología ágil que pone un énfasis particular en la calidad del software y la mejora continua mediante prácticas de desarrollo intensivas. Se enfoca en la colaboración cercana entre los desarrolladores y el cliente, con ciclos de retroalimentación rápidos y frecuentes. XP promueve prácticas como el desarrollo iterativo, la programación en pareja, la integración continua, las pruebas automatizadas y la refactorización constante del código. Su objetivo es entregar software funcional de alta calidad de manera eficiente, respondiendo rápidamente a los cambios y asegurando la satisfacción del cliente mediante la mejora continua en cada iteración [12].

Python

Python es un lenguaje de programación interpretado, de alto nivel y multiparadigma que se ha vuelto extremadamente popular en los últimos años. Creado por Guido van Rossum y lanzado por primera vez en 1991, Python destaca por su sintaxis clara y legible, que favorece la escritura de código limpio y fácil de entender. Una de las filosofías clave de Python es la legibilidad del código, lo que se refleja en su lema "The Zen of Python", que enfatiza la importancia de la claridad y la simplicidad en el diseño del software.

Python es conocido por su versatilidad y puede ser utilizado en una amplia gama de aplicaciones, desde desarrollo web y científico hasta automatización de tareas y desarrollo de juegos. Una de las razones de su popularidad es su amplia biblioteca estándar, que ofrece un conjunto de módulos y funciones predefinidas que facilitan el desarrollo de diversas aplicaciones sin necesidad de escribir código desde cero.

Además, Python es un lenguaje interpretado, lo que significa que el código fuente se ejecuta directamente por un intérprete sin necesidad de compilación previa, lo que agiliza el proceso de desarrollo y depuración. Python también es conocido por su comunidad activa y acogedora, que contribuye constantemente con nuevas bibliotecas, herramientas y recursos educativos que hacen que aprender y trabajar con Python sea aún más atractivo. En resumen, Python es un lenguaje poderoso y versátil que ha ganado popularidad debido a su sintaxis clara y legible, su amplia biblioteca estándar y su comunidad activa y acogedora[13].

Sistema informático

Un sistema informático es un conjunto de componentes interrelacionados que trabajan juntos para procesar, almacenar y transmitir información. Estos componentes incluyen hardware (como computadoras, dispositivos de almacenamiento y periféricos), software (como sistemas operativos y aplicaciones) y personas (usuarios y personal de soporte).

El hardware de un sistema informático incluye la computadora en sí, así como todos los dispositivos periféricos necesarios, como impresoras, escáneres y dispositivos de

almacenamiento. El software de un sistema informático incluye el sistema operativo que controla la computadora, así como las aplicaciones que los usuarios utilizan para realizar tareas específicas, como procesadores de texto, hojas de cálculo y programas de diseño.

Las personas también son una parte importante de un sistema informático, ya que son quienes utilizan el hardware y el software para realizar tareas. El personal de soporte, como los administradores de sistemas, también es crucial para mantener el sistema informático funcionando de manera eficiente y segura.[13]

Eficiencia operativa

La eficiencia operativa se refiere a la capacidad de una organización para utilizar de manera óptima sus recursos y procesos para lograr sus objetivos de manera efectiva. Se trata de maximizar la productividad y minimizar el desperdicio, asegurando que los recursos se utilicen de la mejor manera posible para obtener resultados positivos.

Para lograr la eficiencia operativa, las organizaciones suelen implementar medidas como la mejora de procesos, la automatización de tareas, la optimización de la cadena de suministro, la gestión eficaz de inventarios y la formación y capacitación del personal. También se enfocan en la implementación de tecnologías y sistemas de información que ayuden a mejorar la eficiencia y la toma de decisiones.

La eficiencia operativa es fundamental para el éxito a largo plazo de una organización, ya que puede conducir a una mayor rentabilidad, una mejor calidad de productos y servicios, una mayor satisfacción del cliente y una mayor competitividad en el mercado. Es importante que las organizaciones estén constantemente evaluando y mejorando sus procesos y operaciones para mantenerse eficientes y adaptarse a los cambios en el entorno empresarial[14].

Análisis de datos

El análisis de datos es el proceso de inspeccionar, limpiar, transformar y modelar datos con el objetivo de descubrir información útil, llegar a conclusiones y tomar decisiones informadas. Este proceso se realiza utilizando técnicas y herramientas específicas para

explorar y entender los datos, identificar patrones, tendencias y relaciones, y comunicar los hallazgos de manera efectiva.

Algunas de las técnicas comunes utilizadas en el análisis de datos incluyen[15]:

- **Análisis descriptivo:** Consiste en resumir y describir los datos de manera que sean fácilmente comprensibles. Esto incluye calcular estadísticas como media, mediana, moda, desviación estándar, entre otros, así como la creación de gráficos y visualizaciones para representar los datos de forma visual.

- **Análisis exploratorio:** Se utiliza para investigar los datos en busca de patrones, tendencias, relaciones o anomalías. Esto puede implicar el uso de técnicas como clustering (agrupamiento), análisis de componentes principales, análisis de regresión, entre otros, dependiendo de los datos y el objetivo del análisis.

- **Análisis predictivo:** Se centra en predecir futuros eventos o tendencias basándose en datos históricos. Esto se logra mediante el uso de modelos predictivos, como modelos de regresión, series temporales, o machine learning, que pueden predecir resultados futuros en función de variables pasadas.

- **Análisis prescriptivo:** Se enfoca en recomendar acciones o decisiones a partir de los resultados del análisis. Esto implica el uso de modelos avanzados y algoritmos para generar recomendaciones basadas en datos[15].

Seguridad informática

Garantizar la seguridad de los datos y la información sensible de la panadería. Esto implica implementar medidas de seguridad como encriptación, autenticación de usuarios, auditorías de seguridad y protección contra amenazas cibernéticas como malware y ataques de denegación de servicio[16].

Gestión de productos

La gestión de productos es un proceso integral que abarca el ciclo de vida completo de un producto, desde su concepción hasta su retiro del mercado. Este proceso implica

la planificación, desarrollo, lanzamiento, marketing y venta de productos, así como la gestión de su rendimiento y evolución a lo largo del tiempo.

Mejorar la gestión de productos mediante el análisis de series de tiempo para predecir la demanda, optimizar los niveles de inventario y mejorar la planificación de la producción. Esto permitirá a la panadería reducir costos, evitar la escasez de productos y mejorar la satisfacción del cliente al tener siempre disponible lo que buscan[17].

Series de tiempo

Una serie temporal es un conjunto de observaciones registradas en intervalos de tiempo regulares. El análisis de estas series implica la comprensión de patrones ocultos, tendencias y estacionalidad. En el ámbito de la IA, las series de tiempo se utilizan para predecir eventos futuros a partir de datos históricos. La IA puede ser un multiplicador de fuerza en términos de aumentar la precisión de los pronósticos, ya que tiene la capacidad de estudiar y descubrir patrones fascinantes en los datos automáticamente. Para analizar una serie de tiempo, es importante tener en cuenta varios componentes clave:

- **Tendencia:** Es la dirección general en la que se mueven los datos a largo plazo. Puede ser ascendente, descendente o plana.
- **Estacionalidad**: Son patrones que se repiten a intervalos regulares, como estaciones del año, meses, semanas, días, etc.
- **Ciclos:** Son fluctuaciones no periódicas que se producen en intervalos de tiempo más largos, como ciclos económicos o tendencias políticas.
- **Componente aleatorio:** Es la variabilidad no sistemática o aleatoria que no puede explicarse por la tendencia, la estacionalidad o los ciclos.

El análisis de series de tiempo implica técnicas como suavizado (para identificar tendencias y estacionalidad), descomposición (para separar los componentes de la serie), pronósticos (para predecir futuros valores) y pruebas de estacionariedad (para verificar si la serie es estacionaria, es decir, si su media y varianza son constantes en el tiempo)[18].

1.4 Objetivos

1.4.1 Objetivo general

Implantar un sistema informático aplicando Python para la gestión de productos con series de tiempo en la panadería y pastelería “Flor de Cebada".

1.4.2 Objetivos específicos

- Analizar los procesos de gestión de productos en la panadería y pastelería “Flor de Cebada”.
- Investigar la aplicación de Machine Learning en el uso de series de tiempo en Python.
- Desarrollar un sistema informático usando Python para la gestión de productos en la panadería y pastelería “Flor de Cebada”.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para recolectar información para el presente proyecto se utilizó herramientas como una encuesta para todo el personal de la panadería, una entrevista para la persona encargada de la misma y además de una ficha de observación. Estas herramientas permitieron obtener datos relevantes sobre los procesos actuales, necesidades y desafíos del negocio. La combinación de métodos cualitativos y cuantitativos facilitó una visión integral para el diseño del sistema propuesto.

A continuación, se muestra el modelo de la encuesta aplicado al personal de la panadería.

a) Encuesta para Trabajadores de la Panadería y Pastelería "Flor de Cebada"

Objetivo: Evaluar las necesidades y opiniones de los trabajadores sobre el proceso actual de gestión de productos en la panadería y pastelería "Flor de Cebada".

Pregunta 1: ¿Con qué frecuencia el trabajador revisa cuántos productos hay disponibles (pan, pasteles, etc.)?

- Varias veces al día.
- Todos los días.
- Una vez por semana.
- Una vez al mes.

Pregunta 2: ¿Cómo se siente el trabajador con la forma actual de manejar los productos en la panadería?

- Muy contento.
- Contento.

- Insatisfecho.
- Muy insatisfecho.

Pregunta 3: ¿Qué problemas encuentra el trabajador más a menudo al gestionar los productos?

- No encuentra problemas al gestionar productos.
- Es difícil saber cuántos productos se han vendido o están disponibles.
- Tiene dificultades para contar los productos.
- No hay control sobre cuándo caducan los productos.

Pregunta 4: ¿Cree el trabajador que un sistema que automatice la gestión de productos le facilitaría el trabajo?

- Sí.
- No.

Pregunta 5: ¿Ha usado el trabajador alguna vez un sistema o herramienta que le ayude a llevar el control de productos?

- Sí.
- No.

Pregunta 6: ¿Qué tan cómodo se sentiría el trabajador usando una computadora o una herramienta nueva para gestionar los productos?

- Muy cómodo.
- Algo cómodo.
- Poco cómodo.
- No se sentiría cómodo.

Pregunta 7: ¿Qué le gustaría al trabajador que pudiera hacer un nuevo sistema de gestión de productos? (Puede escoger más de una opción)

- Llevar el registro de los productos automáticamente.
- Avisar cuándo un producto va a caducar.
- Ayudar a predecir cuántos productos se necesitarán en los próximos días.
- Generar reportes automáticos sobre ventas.

Pregunta 8: ¿Con qué frecuencia el trabajador tiene problemas con productos que se terminan o sobran?

- Muchas veces.
- A veces.
- Casi nunca.
- Nunca.

Pregunta 9: Cuando un producto está cerca de caducar, ¿qué hace el trabajador?

- Lo revisa manualmente.
- Usa algún tipo de sistema para revisarlo.
- No lleva un control específico.

Pregunta 10: ¿Qué tan importante cree el trabajador que es poder predecir cuántos productos se necesitarán en el futuro?

- Muy importante.
- Importante.
- Poco importante.
- No es importante.

- A continuación, se muestra una entrevista al jefe de la panadería.

Guía de Entrevista para el jefe de la Panadería y Pastelería "Flor de Cebada"

Objetivo de la entrevista: Obtener información más profunda sobre las experiencias de los trabajadores en la gestión de productos y su percepción sobre la implementación de un sistema informático.

1. ¿Describiría un día típico en tu trabajo dentro de la panadería?
2. ¿Cómo ha cambiado la forma en que manejan los productos a en la panadería?
3. ¿Cómo llevan el control de la cantidad de productos disponibles (pan, pasteles, otros)?
4. ¿Cómo deciden qué cantidad de productos deben preparar o reponer diariamente?
5. ¿Si tuviera acceso a una herramienta tecnológica que le ayude a gestionar mejor los productos, ¿qué aspectos cree que debería incluir?
6. ¿Qué tipo de información le gustaría que el sistema le proporcionara sobre los productos que maneja?
7. ¿Hay algo que le gustaría agregar o alguna sugerencia que crea importante para que el sistema informático sea exitoso?

A continuación, se muestra la guía de ficha de observación, la cual permite registrar los aspectos clave durante la observación directa de los procesos en la panadería. Esta herramienta facilita la recolección de información detallada sobre las operaciones diarias, brindando datos útiles para el análisis posterior.

Ficha de observación - Análisis de los procesos de gestión de productos

Objetivo: Registrar las actividades relacionadas con la gestión de productos y evaluar la eficiencia de los procesos actuales.

Tabla 4. Modelo de Ficha de observación.

Criterio de Observación	**Descripción**	**Frecuencia**	**Comentarios**
Actualización del inventario	¿Cada cuánto se actualiza el inventario?	Diaria/Semanal/Mensual	
Registro de caducidad	¿Hay control sobre los productos próximos a caducar?	Sí/No	
Uso de herramientas tecnológicas	¿Se usan herramientas tecnológicas en el proceso?	Frecuente/Raro	
Gestión de productos con baja demanda	¿Cómo se gestionan los productos de baja demanda?	Manual/Automatizada	
Precisión en la cantidad de productos	¿Se registra correctamente la cantidad de productos vendidos?	Alta/Baja	

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación se contextualizó en la modalidad de investigación de campo y bibliográfica-documental.

2.2.2 Población y muestra

La población con la cual se trabajó en el proyecto corresponde al personal de la panadería y pastelería 'Flor de Cebada'. En la Tabla 5, se proporciona un desglose

detallado de los diversos roles desempeñados por el personal, incluyendo las funciones que abarcan la totalidad de la panadería. Este análisis permitió identificar las responsabilidades clave y su relación con la gestión de productos, facilitando así la implementación del sistema propuesto.

Tabla 5. Población de estudio.

Población	Número	Porcentaje
Panadero	1	25 %
Pastelero	1	25 %
Horneador	1	25 %
Cobrador	1	25 %
Total	4	100%

En esta investigación, se trabajará de manera integral con la totalidad de una población que no supera los cien elementos, evitando así la necesidad de seleccionar muestras representativas.

2.2.3 Recolección de información

Validación del Instrumento Alfa de Cronbach en la encuesta dirigida al personal de la panadería y pastelería "Flor de Cebada"

El coeficiente de Alfa de Cronbach es una medida estadística empleada para evaluar la confiabilidad o consistencia interna de un conjunto de ítems en una prueba o encuesta. Su valor oscila entre 0 y 1, donde un valor más alto indica mayor consistencia interna. Se utiliza comúnmente la Escala de Likert para estructurar las preguntas, lo que ayuda a asegurarse de que los ítems sean consistentes y fiables al medir el mismo constructo o variable. Este coeficiente es crucial para garantizar que los resultados obtenidos en una encuesta o prueba sean representativos y estables, lo que aumenta la validez de los análisis realizados con los datos recogidos.

$$a = \frac{k}{k-1}\left[1 - \frac{\sum v_i}{vt}\right]$$

Tabla 6. Cálculo y sumatoria de varianza de la encuesta.

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 6	Pregunta 8	Pregunta 10	Suma
Sujeto 1	2	1	3	1	2	1	8
Sujeto 2	2	2	4	2	3	1	12
Sujeto 3	3	2	4	2	3	2	13
Sujeto 4	2	2	1	1	2	1	7
Varianza	0,1875	0,1875	1,5	0,25	0,25	0,1875	
Sumatoria de varianza de los items	2,375						
Varianza total del instrumento	6,5						

Tabla 7. Cálculo de Alfa de Cronbach.

a	coeficiente de confiabilidad	0,84615387
K	Numero de items de instrumento	4
$\sum v_i$	Sumatoria de varianza de los items	2.735
vt	Varianza total del instrumento	6.5

Tabla 8. Tabla de confiabilidad de Alfa de Cronbach.

Alfa de Cronbach	Interpretación
0,9 - 1	Excelente
0,8 - 0,9	Buena
0,7 - 0,8	Aceptable
0,6 - 0,7	Débil
0,5 - 0,6	Pobre
< 0,5	Inaceptable

Con el resultado de Alfa de Cronbach con un valor de 0,84615387 tiene un nivel bueno de confiabilidad.

Encuesta realizada

Se realizó una encuesta a los empleados de la panadería y pastelería 'Flor de Cebada', cuyo propósito fue recopilar información sobre los procesos actuales de gestión. Las preguntas de la encuesta se diseñaron para abarcar aspectos clave como la organización

de inventarios, los métodos utilizados para el registro de productos y las principales dificultades enfrentadas en el flujo operativo diario. Además, se buscó identificar las necesidades específicas del personal en relación con un sistema informático que facilite y optimice la gestión de productos. Los resultados obtenidos fueron fundamentales para orientar el diseño del sistema, garantizando que responda a las exigencias y características propias del negocio.

Pregunta 1: ¿Con qué frecuencia el trabajador revisa cuántos productos hay disponibles (pan, pasteles, etc.)?

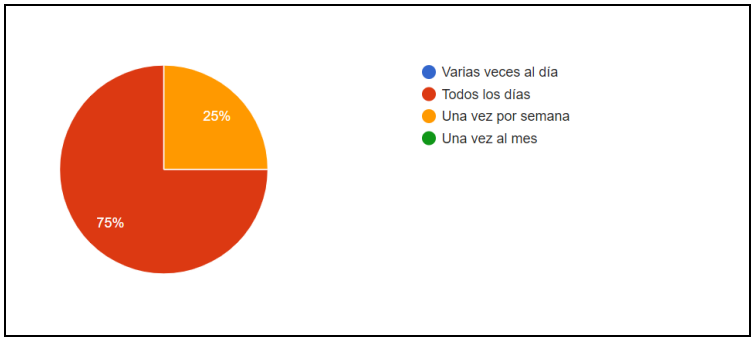


Figura 1. Frecuencia con la que el trabajador revisa la disponibilidad de los productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 1** muestra que la mayoría de los trabajadores (75%) revisa diariamente la disponibilidad de productos como pan y pasteles, lo que indica un alto nivel de monitoreo y control diario sobre los inventarios. Sin embargo, el 25% de los encuestados realiza esta revisión solo una vez por semana, lo que sugiere que podría haber algunas diferencias en los hábitos o responsabilidades de los empleados respecto al control de inventarios. Esto podría influir en la precisión de la gestión de productos y podría ser un aspecto a mejorar para lograr una mayor consistencia entre los trabajadores.

Pregunta 2: ¿Cómo se siente el trabajador con la forma actual de manejar los productos en la panadería?

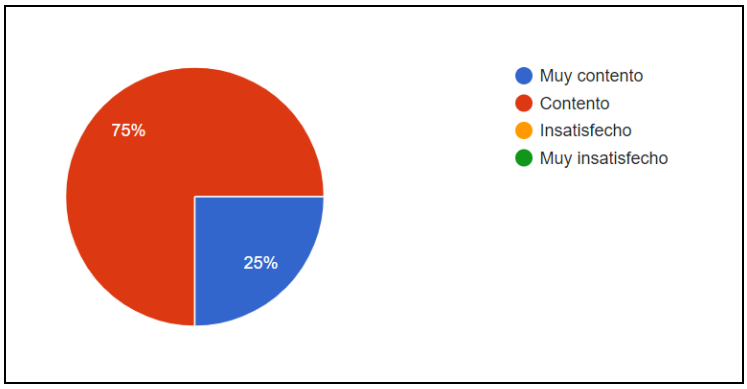


Figura 2. Forma actual de manejo de los productos de la panadería

Análisis e interpretación de los resultados

El análisis de la **Figura 2** revela que la mayoría de los trabajadores (75%) se siente "contento" con la forma actual de manejar los productos en la panadería, mientras que el 25% se siente "muy contento." Estos resultados indican una percepción generalmente positiva del sistema de gestión de productos. No se reportaron niveles de insatisfacción, lo que sugiere que el sistema actual cumple con las expectativas de los empleados, aunque podría explorarse la posibilidad de mejoras para elevar aún más la satisfacción laboral.

Pregunta 3: ¿Qué problemas encuentra el trabajador más a menudo al gestionar los productos?

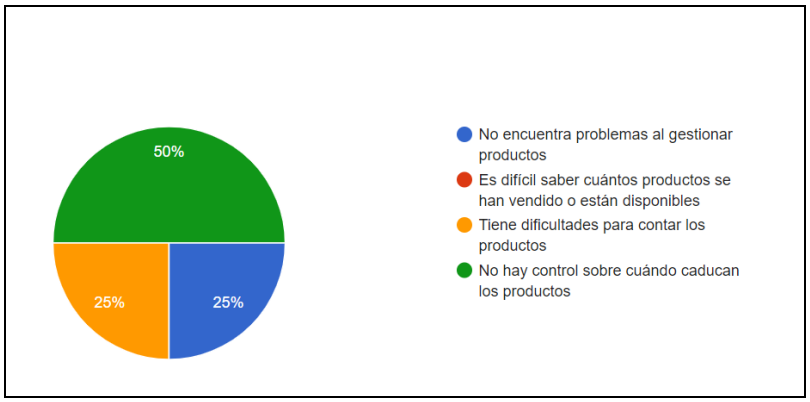


Figura 3. Problemas que encuentra el trabajador con el manejo de los productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 3** muestra que el 50% de las respuestas no tienen un control sobre cuando caducan los productos, el 25% de los trabajadores no encuentra dificultades para contar los productos, mientras que el 25% restante señala que es complicado saber cuántos productos se han vendido o están disponibles. Estos resultados destacan que la mayor parte de los problemas percibidos están relacionados con el control del inventario, lo que sugiere la necesidad de implementar soluciones que faciliten el conteo de productos y el seguimiento de ventas para mejorar la eficiencia en la gestión de productos.

Pregunta 4: ¿Cree el trabajador que un sistema que automatice la gestión de productos le facilitaría el trabajo?

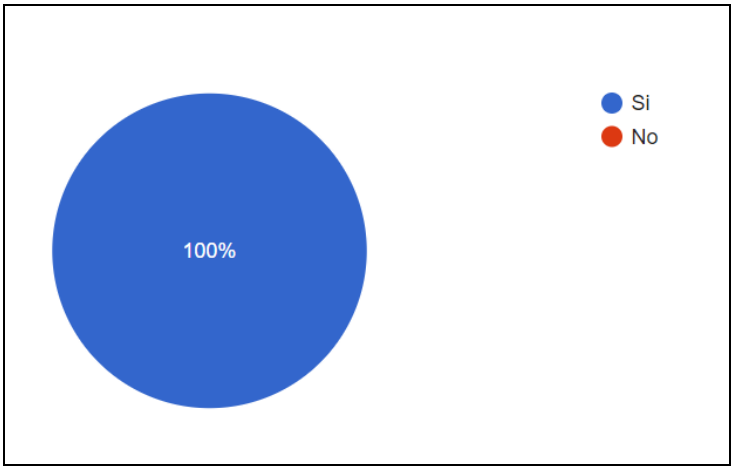


Figura 4. La automatización del sistema ayudaría al trabajo.

Análisis e interpretación de los resultados

El análisis de la **Figura 4** revela que el 100% de los trabajadores cree que un sistema que automatice la gestión de productos les facilitaría el trabajo. Este resultado refleja un fuerte consenso a favor de la implementación de soluciones tecnológicas que optimicen el proceso de gestión, lo que sugiere que la automatización podría mejorar significativamente la eficiencia y reducir las dificultades actuales en el manejo de productos.

Pregunta 5: ¿Ha usado el trabajador alguna vez un sistema o herramienta que le ayude a llevar el control de productos?

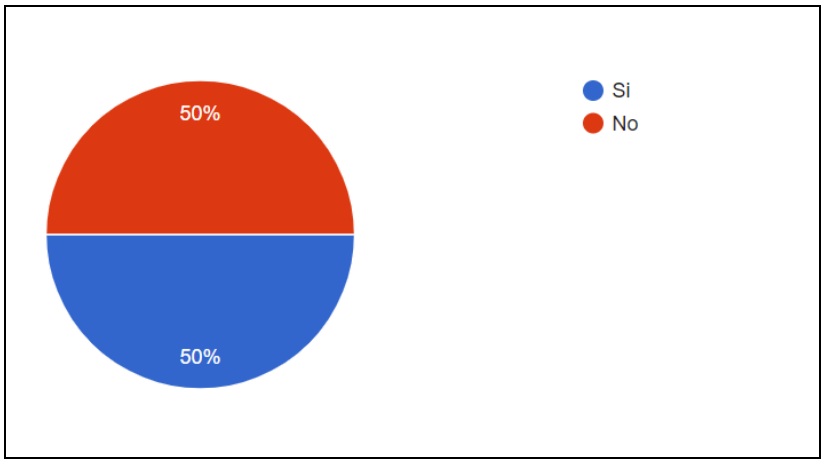


Figura 5. Ha usado una herramienta de gestión de productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 5** muestra que el 50% de los trabajadores ha utilizado algún sistema o herramienta para el control de productos, mientras que el otro 50% no lo ha hecho. Estos resultados indican una división equitativa en la experiencia previa con sistemas de gestión, lo que sugiere que una parte significativa de los empleados podría requerir capacitación o adaptación al implementar un sistema automatizado en la panadería.

Pregunta 6: ¿Qué tan cómodo se sentiría el trabajador usando una computadora o una herramienta nueva para gestionar los productos?

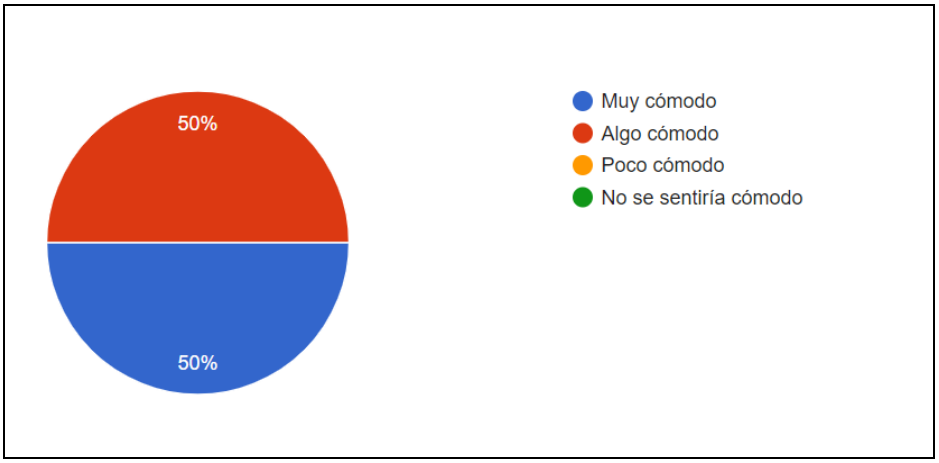


Figura 6. Comodidad del trabajado al usar una herramienta de gestión de productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 6** revela que el 50% de los trabajadores se sentiría "muy cómodo" usando una computadora o una nueva herramienta para gestionar los productos, mientras que el otro 50% se sentiría "algo cómodo." Estos resultados reflejan una disposición positiva hacia el uso de tecnología en la gestión de productos, lo que sugiere que la mayoría de los empleados estaría abierta a la implementación de un sistema informático, aunque podría ser beneficioso ofrecer soporte adicional para asegurar una transición fluida para todos.

Pregunta 7: ¿Qué le gustaría al trabajador que pudiera hacer un nuevo sistema de gestión de productos? (Puede escoger más de una opción)

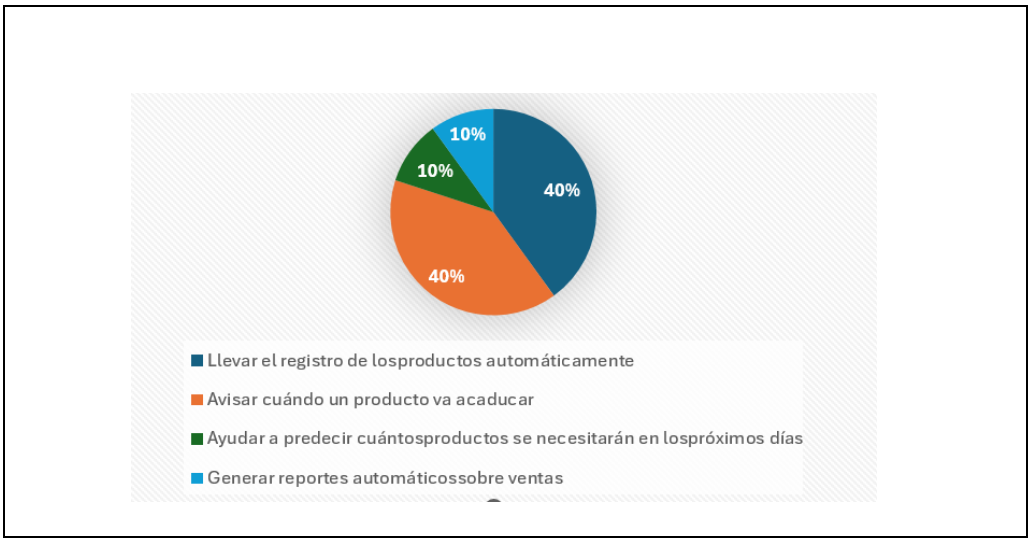


Figura 7. Opciones que le gustaría al trabajador para la gestión de productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 7** revela que todos los trabajadores valoran que un nuevo sistema de gestión de productos automatice el registro de productos (100%) y avise cuándo un producto está próximo a caducar (100%). Sin embargo, solo el 25% expresó interés en que el sistema ayude a predecir la demanda futura de productos, y otro 25% indicó que le gustaría que generara reportes automáticos sobre ventas. Estos resultados sugieren que las funciones relacionadas con el control de inventario y fechas de caducidad son las más prioritarias para los trabajadores, mientras que las capacidades avanzadas como la predicción de necesidades y generación de reportes podrían tener menos urgencia inicial, pero aún serían útiles para algunos.

Pregunta 8: ¿Con qué frecuencia el trabajador tiene problemas con productos que se terminan o sobran?

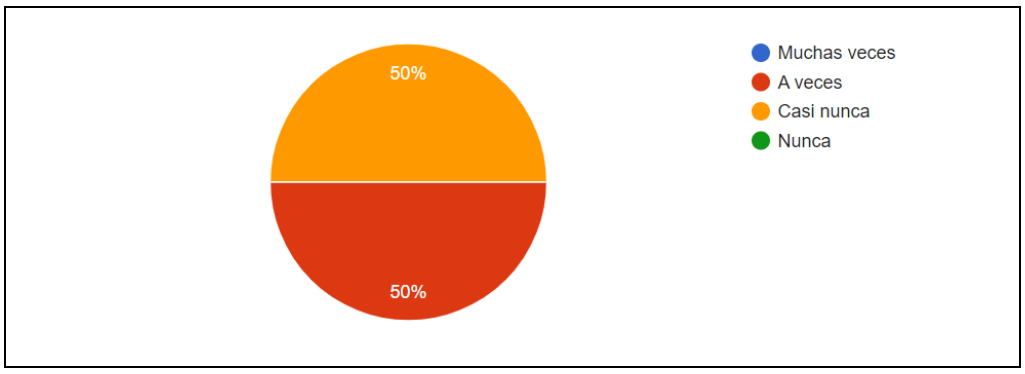


Figura 8. Frecuencia con la que el trabajador tiene problemas con productos.

Análisis e interpretación de los resultados

El análisis de la **Figura 8** muestra que el 50% de los trabajadores tiene problemas con productos que se terminan o sobran "a veces," mientras que el otro 50% indicó que esto ocurre "casi nunca." Estos resultados sugieren que, aunque la mitad de los empleados enfrenta ocasionalmente dificultades con el manejo de inventarios, la otra mitad percibe un control más estable. Esto indica que, aunque no es un problema generalizado, aún existe margen para mejorar la precisión en la gestión de inventarios y reducir los casos en los que se generan excesos o faltantes.

Pregunta 9: Cuando un producto está cerca de caducar, ¿qué hace el trabajador?

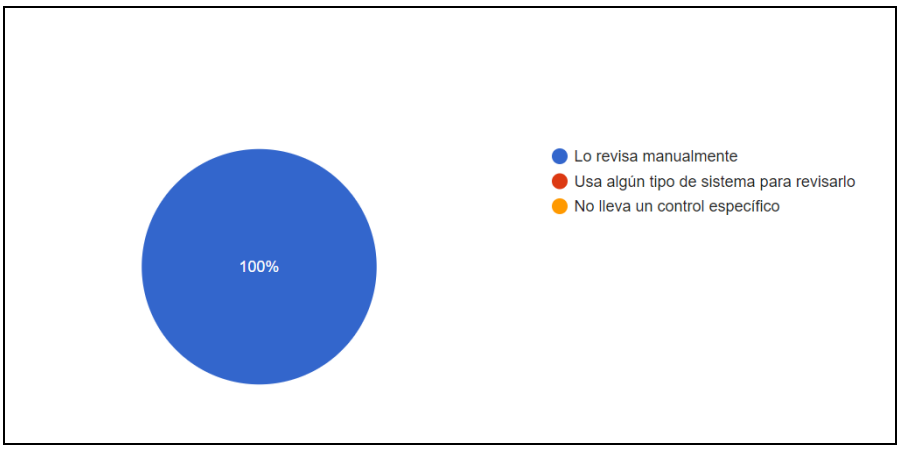


Figura 9. Las consecuencias de un producto vencido en la panadería..

Análisis e interpretación de los resultados.

El análisis de la **Figura 9** revela que el 100% de los trabajadores revisa manualmente los productos que están cerca de caducar. Este resultado indica que no se está utilizando ningún sistema automatizado para el control de caducidades, lo que podría incrementar el riesgo de desperdicio de productos. La dependencia de revisiones manuales sugiere la necesidad de implementar un sistema que facilite este seguimiento, optimizando la gestión de inventarios y ayudando a prevenir pérdidas debido a productos caducados.

Pregunta 10: ¿Qué tan importante cree el trabajador que es poder predecir cuántos productos se necesitarán en el futuro?

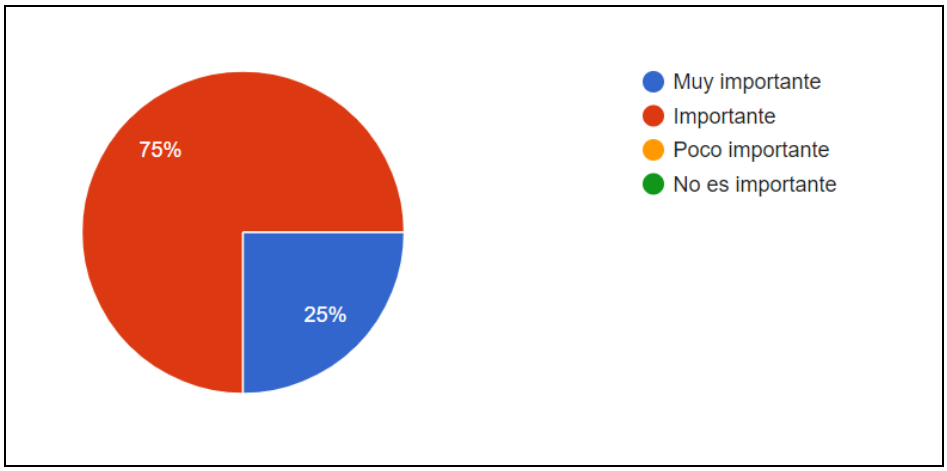


Figura 10. Importancia de los productos de los datos en el futuro.

Análisis e interpretación de los resultados

El análisis de la **Figura 10** muestra que el 75% de los trabajadores considera "importante" poder predecir cuántos productos se necesitarán en el futuro, mientras que el 25% lo califica como "muy importante." Estos resultados indican que la mayoría de los empleados reconoce la relevancia de las previsiones en la gestión de inventarios, lo que sugiere que la implementación de un sistema que permita realizar predicciones podría ser altamente beneficiosa. Esta capacidad no solo ayudaría a optimizar la planificación de la producción, sino que también podría reducir la acumulación de excedentes y mejorar la eficiencia operativa en la panadería.

Entrevista:

Se realizó una encuesta dirigida a la encargada de la panadería y pastelería Flor de Cebada, con el fin de obtener información detallada sobre la gestión actual de los procesos y el uso de recursos tecnológicos.

1. ¿Describiría un día típico en tu trabajo dentro de la panadería?

En un día típico, llego temprano y comienzo revisando los productos disponibles. Luego, paso a preparar lo que falta, como el pan y los pasteles. Organizamos los productos en las vitrinas y hacemos un seguimiento de las ventas a lo largo del día. También reviso si es necesario reponer o preparar más productos en función de la demanda.

Observación: La rutina parece bien establecida, pero depender exclusivamente de la revisión manual puede generar problemas en la precisión de la gestión de inventario.

2. ¿Cómo ha cambiado la forma en que manejan los productos en la panadería?

Con el tiempo, hemos mejorado algunos procesos, como hacer un seguimiento más riguroso de los productos. Sin embargo, sigue siendo manual y dependemos de nuestra experiencia y observación para la toma de decisiones.

Observación: No se mencionan mejoras tecnológicas significativas, lo que indica una posible área de mejora.

3. ¿Cómo llevan el control de la cantidad de productos disponibles (pan, pasteles, otros)?

El control se realiza de manera visual y manual. Contamos los productos diariamente, al finalizar la jornada, y verificamos lo que queda para decidir si se debe reponer.

Observación: Un sistema automatizado podría reducir errores y mejorar la eficiencia del control.

4. ¿Cómo deciden qué cantidad de productos deben preparar o reponer diariamente?

La decisión se basa en la experiencia y en el comportamiento de las ventas diarias. También tomamos en cuenta si es un día regular o de alta demanda, como los fines de semana.

Observación: Un sistema basado en datos históricos podría optimizar la producción y minimizar desperdicios.

5. ¿Si tuviera acceso a una herramienta tecnológica que le ayude a gestionar mejor los productos, ¿qué aspectos cree que debería incluir?

Me gustaría que el sistema pudiera llevar un registro automático de los productos que se venden y los que quedan en la panadería.

6. ¿Qué tipo de información le gustaría que el sistema le proporcionara sobre los productos que maneja?

Me gustaría recibir información precisa sobre los productos disponibles, productos próximos a caducar, y la información de las ventas.

7. ¿Hay algo que le gustaría agregar o alguna sugerencia que crea importante para que el sistema informático sea exitoso?

El sistema debe ser fácil de usar, considerando que no todos los trabajadores tienen familiaridad con la tecnología.

Conclusión de la entrevista

La entrevista revela que, aunque la panadería ha establecido una rutina de trabajo eficiente, la gestión actual de productos se basa principalmente en procesos manuales y la experiencia de los trabajadores. La dependencia de métodos visuales y manuales para el control de inventarios y la toma de decisiones sobre la producción muestra una falta de integración de tecnologías que podrían optimizar estos procesos. Los entrevistados identificaron la necesidad de un sistema automatizado que no solo registre la cantidad de productos vendidos y disponibles, sino que también ofrezca información sobre fechas de caducidad y patrones de ventas. La implementación de un sistema tecnológico intuitivo podría mejorar la eficiencia operativa, reducir errores en el manejo de inventarios y facilitar la toma de decisiones basadas en datos históricos.

Además, es fundamental considerar la capacitación de los empleados para asegurar la correcta adopción de cualquier nueva herramienta, garantizando así su éxito y eficacia en la gestión de productos dentro de la panadería.

Ficha de Observación

Se llevó a cabo una ficha de observación con el propósito de analizar de manera directa los procesos operativos y la interacción del personal con las herramientas y métodos actuales de gestión. Este enfoque permitió identificar puntos críticos, ineficiencias y áreas de mejora que no siempre son evidentes a través de otros métodos de recolección de datos, proporcionando una visión integral de las operaciones diarias.

Tabla 9. Resultado de la ficha de observación.

Criterio de Observación	**Descripción**	**Frecuencia**	**Comentarios**
Actualización del inventario	¿Cada cuánto se actualiza el inventario?	Diaria	La actualización del inventario se realiza manualmente al final de cada jornada, lo que implica riesgos de errores humanos.
Registro de caducidad	¿Hay control sobre los productos próximos a caducar?	No	Actualmente no hay un sistema específico para controlar la caducidad, se basa en la observación manual.
Uso de herramientas tecnológicas	¿Se usan herramientas tecnológicas en el proceso?	Raro	El uso de tecnología es limitado; la mayoría de los procesos se manejan manualmente.
Gestión de productos con baja demanda	¿Cómo se gestionan los productos de baja demanda?	Manual	No hay una estrategia formal; la producción de estos productos se reduce en función de la observación y experiencia.
Precisión en la cantidad de productos	¿Se registra correctamente la cantidad de productos vendidos?	Baja	Al no tener un sistema no se registran las ventas a excepción de las facturas o todo se hace manual.

Conclusión de la ficha de Observación:

La gestión de productos en la panadería "Flor de Cebada" se realiza mayormente de manera manual, lo que genera ineficiencias y errores en el control del inventario y en la precisión de los productos vendidos. La falta de herramientas tecnológicas adecuadas y un sistema para el registro de caducidad representa un riesgo para la optimización del proceso, aumentando la posibilidad de desperdicios y pérdidas. Implementar un sistema automatizado permitiría mejorar la precisión en el manejo de productos, facilitar el control de inventarios, y gestionar de manera más eficiente los productos de baja demanda.

2.2.4 Procesamiento y análisis de datos

De acuerdo con la información recolectada de las encuestas, entrevistas y la ficha de observación, se establece el siguiente análisis:

- Es necesario implementar un sistema automatizado para la gestión de productos en la panadería, ya que la actual dependencia de métodos manuales genera ineficiencias y errores en el control de inventarios.

- Aunque la satisfacción laboral es generalmente positiva, hay margen para mejorar el sistema de gestión de productos para elevar aún más la percepción de los trabajadores y generar resultados positivos.

- Un alto porcentaje de los empleados enfrenta dificultades al contar los productos, lo que resalta la urgencia de soluciones que faciliten el seguimiento de inventarios y ventas.

- La totalidad de los trabajadores está a favor de la automatización en la gestión de productos, destacando que un sistema que registre automáticamente los productos y avise sobre fechas de caducidad podría mejorar significativamente la eficiencia operativa.

- La experiencia previa con herramientas de control de productos es equitativa entre los empleados, indicando que es fundamental ofrecer capacitación para asegurar una correcta adaptación al nuevo sistema.

- Aunque la mitad de los trabajadores se sienten cómodos utilizando tecnología, es esencial brindar soporte adicional para facilitar la transición a un sistema automatizado, asegurando así una adopción exitosa.

- La mayoría de los empleados reconoce la importancia de prever la demanda futura, lo que subraya la necesidad de implementar un sistema que permita realizar estas predicciones para mejorar la planificación y la eficiencia operativa en la panadería.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

En el capítulo II del trabajo de investigación, se llevó a cabo la recopilación de datos mediante diversas técnicas y herramientas de recolección. A partir de la información obtenida, se identificaron varias necesidades que pueden ser resueltas.

3.1.1 Análisis de gestión de procesos

Haciendo un análisis de gestión de procesos, se elaboraron dos diagramas: uno del flujo actual de gestión de productos y ventas, y otro optimizado según las necesidades identificadas. Estos diagramas permiten comparar el estado actual con la propuesta de mejora, facilitando la identificación de áreas de optimización.

a) Proceso actual

Se elaboró un diagrama de procesos basado en el análisis de la gestión actual de productos y ventas en la empresa. Este diagrama refleja cómo se llevan a cabo dichos procesos en la situación actual, permitiendo una comprensión clara de su funcionamiento.

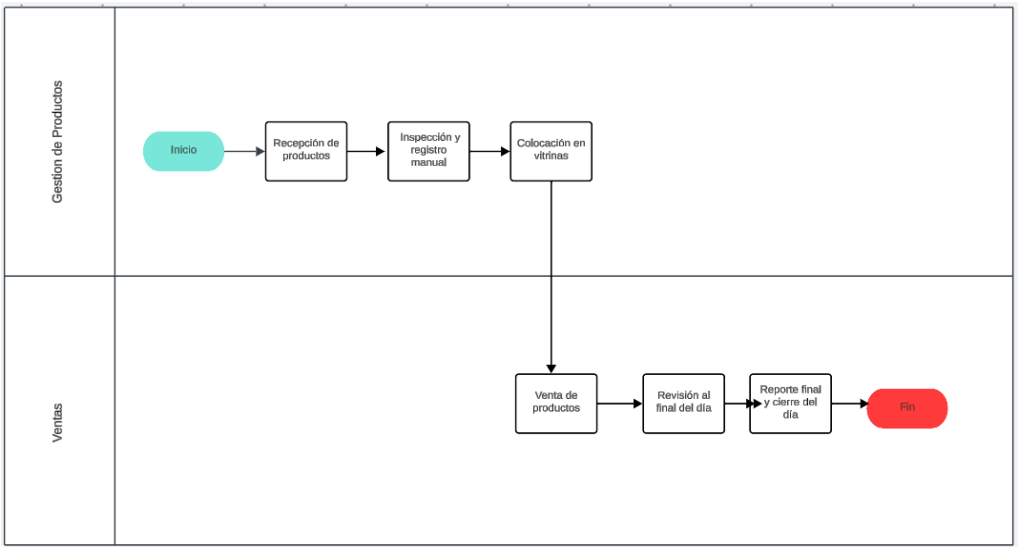


Figura 11. Análisis de procesos actual.

El diagrama de flujo de trabajo presentado en la **Figura 11** las actividades diarias en la gestión de productos de la panadería y pastelería “Flor de Cebada”.

Inicio: El proceso comienza con la preparación para recibir los productos.

Recepción de productos: Se lleva a cabo la recepción de los artículos que serán gestionados y vendidos.

Inspección y registro manual: Cada producto es inspeccionado y registrado manualmente para garantizar su calidad y trazabilidad.

Colocación en vitrinas/estanterías: Los productos se organizan adecuadamente en las vitrinas o estanterías para su exhibición.

Venta de productos: Se realiza la venta directa de los productos a los clientes.

Anotación de ventas: Cada venta es registrada manualmente para llevar un control detallado.

Revisión al final del día: Al finalizar la jornada, se realiza una revisión de los productos vendidos y sobrantes.

Sumatoria manual de ganancias y sobrantes: Se calcula manualmente el total de ganancias y los productos restantes para verificar la concordancia.

Reporte final y cierre del día: Se elabora un reporte final con los datos recopilados, lo que permite cerrar oficialmente las actividades del día.

Fin: El flujo termina con el cierre del día, dejando todo preparado para reiniciar el ciclo al día siguiente.

b) Proceso automatizado

El proceso automatizado se ha diseñado a partir del análisis del proceso actual de gestión de productos y ventas en la empresa. Este nuevo flujo busca mejorar la eficiencia y eficacia de las operaciones, incorporando soluciones a las necesidades identificadas mediante las herramientas de recolección de datos.

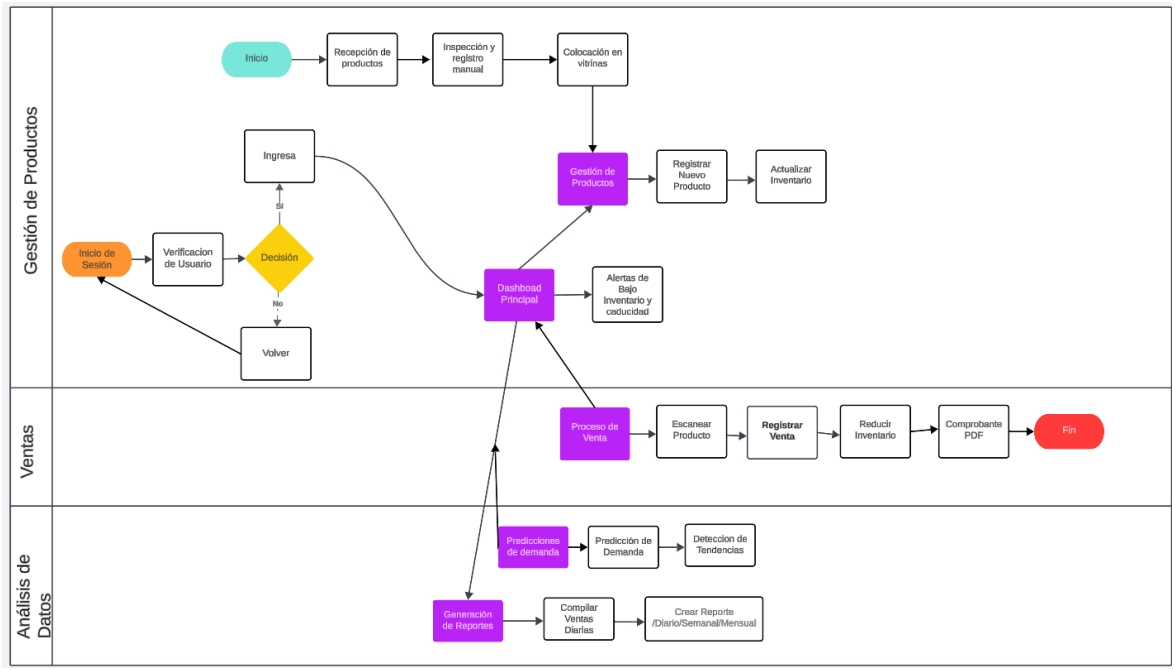


Figura 12. análisis de procesos actualizado.

Gestión de productos

Inicio de Sesión: Este es el primer paso, donde se da inicio al proceso.

- **Verificación de usuario:** Una vez que el usuario inicia sesión, se valida su autenticidad para permitir el acceso a las siguientes etapas.
- **Decisión (rombo):** En este punto, se toma una decisión que determina el flujo siguiente del proceso.
- **Valor:** Se realiza una evaluación o revisión relacionada con los productos.
- **Dashboard principal:** Si se pasa la decisión, se llega a la interfaz principal del sistema.
- **Generación de productos:** Desde el dashboard, el sistema genera productos disponibles.
- **Registrar Nuevo producto:** Aquí se añaden nuevos productos al sistema.
- **Actualizar inventario:** Esta etapa permite realizar actualizaciones en el inventario de productos.

- **Recepción de productos:** Este proceso cubre la recepción de productos que llegan a la panadería.

- **Inspección y registro manual:** Se lleva a cabo una inspección y registro manual de los productos recibidos.

- **Colocación en vitrinas**: Los productos aprobados se colocan en las vitrinas disponibles para la venta.

Ventas

- **Proceso de ventas:** Esta sección inicia con el proceso de ventas desde el dashboard principal.

- **Escanear producto:** Se escanea el código del producto para su identificación en el sistema.

- **Registrar venta:** El producto escaneado se registra como parte de una venta.

- **Reducir inventario:** Tras registrar la venta, el inventario disponible se reduce para reflejar la venta realizada.

- **Comprar PDF:** El cliente recibe un recibo en formato PDF por la compra realizada.

- **Fin (rojo):** El proceso de ventas finaliza con este paso, señalando el término del flujo.

Análisis de datos

- **Predicción de demanda:** A partir de los datos disponibles, se realizan predicciones sobre la demanda futura de productos.

- **Detección de tendencias:** Se analizan las tendencias de consumo para anticipar patrones y necesidades.

- **Crear reportes diarios/semanales/mensuales:** El sistema genera reportes que detallan las ventas y el comportamiento del mercado en diferentes intervalos de tiempo.

- **Generación de reportes:** Aquí se elaboran los informes clave para la gestión.

- **Consultar ventas diarias:** Se accede a los reportes diarios para analizar el rendimiento de ventas de un día específico.

3.1.2 Aplicación de machine learning en el uso de series de tiempo

a) Revisión de estudios sobre series de tiempo

Para seleccionar las mejores prácticas en la aplicación de Machine Learning en series de tiempo dentro de un sistema de gestión de productos, a continuación, se presentan tablas comparativas que sintetizan la información de cuatro artículos científicos relevantes, proporcionando una visión detallada de las técnicas, algoritmos y resultados utilizados en proyectos similares. Esta comparación servirá como referencia para identificar los modelos y herramientas más adecuados para optimizar la gestión de productos en la panadería y pastelería “Flor de Cebada”.

Además, se destacarán las ventajas y limitaciones de cada enfoque, lo que permitirá seleccionar aquellos métodos que mejor se adapten a las necesidades específicas del negocio. Este análisis comparativo contribuirá a la toma de decisiones informadas, facilitando la implementación de un sistema de gestión eficiente y preciso para mejorar tanto la planificación de producción como la predicción de demanda. Asimismo, se evaluarán los posibles impactos de cada técnica en la optimización de recursos y reducción de costos operativos, brindando una perspectiva integral sobre la viabilidad de cada modelo en el contexto de la panadería.

Adicionalmente, se examinarán distintos enfoques utilizados en estudios previos sobre la gestión de inventarios mediante técnicas de aprendizaje automático. La comparación se centrará en factores clave como la precisión de las predicciones, la capacidad de adaptación a cambios en la demanda y el impacto en la rentabilidad del negocio. Este análisis permitirá identificar las estrategias más efectivas para optimizar la gestión de

productos en la panadería, enfocándose en la reducción de costos operativos y la mejora en la planificación de compras y ventas.

Por último, se analizarán estudios previos que han implementado estas técnicas en entornos similares, evaluando su éxito y posibles desafíos encontrados durante su aplicación. Se pondrá énfasis en la escalabilidad de cada solución, considerando el crecimiento potencial de la panadería y la necesidad de ajustes continuos en el sistema. La identificación de patrones en los datos históricos permitirá mejorar la toma de decisiones estratégicas, asegurando un abastecimiento adecuado y reduciendo el desperdicio de productos.

Tabla 9. Análisis del paper **"**Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization"[19].

Aspecto	**Descripción**
Titulo	Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization.
Autores	Khalil Nami1, Hassan Labriji. EI Habib Ben Lahmar
Modelos Utilizados	Combina el modelo ARIMA-NARNN (red neuronal autoregresiva no lineal) para predicción de precios de materias primas, XGBoost para pronóstico de demanda, y un modelo de optimización combinatoria para decidir entre vender, comprar o mantener inventario.
Aplicaciones	Útil para cualquier organización que gestione inventarios de productos básicos y busque maximizar beneficios reduciendo tanto costos de almacenamiento como riesgos de desabastecimiento.
Resultados	El modelo propuesto aumenta el rendimiento de inventarios al sugerir cantidades y momentos óptimos de compra y venta. Se logró un incremento de 1.3% en el beneficio anual en comparación con la política de inventarios existente en la empresa de prueba.

Tabla 10. Análisis del paper **"**Inventory Management and Cost Reduction of Supply Chain Processes Using A-Based Time-Series Forecasting and ANN Modeling"[20].

Aspecto	**Descripción**
Titulo	Inventory Management and Cost Reduction of Supply Chain Processes Using A-Based Time-Series Forecasting and ANN Modeling
Autores	Umamaheswaran, Praveen; Ganjeizadeh, Farnaz; Ghasib, Hatim
Modelos Utilizados	Se utiliza modelado de Redes Neuronales Artificiales (ANN) para prever la demanda futura, combinado con un enfoque de pronóstico basado en series de tiempo A. Incluye un algoritmo con una red neuronal de alimentación hacia adelante y retropropagación para ajustar pesos y minimizar el error de predicción.
Aplicaciones	Dirigido a organizaciones de la cadena de suministro que buscan mejorar la gestión de inventarios y reducir costos, optimizando la planificación de recursos y el flujo de productos.
Resultados	Se logró una precisión de pronóstico de 75-80%, mejorando la precisión promedio de 72% de la mayoría de las redes de suministro.

Tabla 11. Análisis del paper **"**Optimization of inventory management through computer vision and machine learning technologies"[21].

Aspecto	**Descripción**
Titulo	Optimization of inventory management through computer vision and machine learning technologies
Autores	William Villegas-Cha, Alexandra Maldonado Navarro, Santiago Sanchez-Viteri
Modelos Utilizados	Comparación de arquitecturas de redes neuronales (AlexNet, VGG16, ResNet, EfficientNet, MobileNet) en términos de precisión, velocidad de inferencia, uso de memoria y capacidad de generalización.
Aplicaciones	Gestión de inventarios en procesos industriales, optimización de la eficiencia operativa y mantenimiento predictivo
Resultados	• Reducción del 45% en el tiempo de conteo de inventario. • Aumento del 9% en la precisión del inventario. • Mejora en la eficiencia operativa.

Tabla 12. Análisis del paper **"**Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis"[22].

Aspecto	**Descripción**
Titulo	Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis
Autores	Sandeep Kumar Panda, Sachi Nandan Mohanty
Modelos Utilizados	Random Forest Regressor, AutoRegressive Integrated Moving Average (ARIMA)Gradient Boosting Regressor (GBR), Light Gradient Boosting Machine Regressor (LightGBM), Extreme Gradient Boosting Regressor (XGBoost), Cat Boost Regressor, Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM
Aplicaciones	Pronóstico de la demanda de alimentos para mejorar la gestión del inventario y reducir el desperdicio en la industria alimentaria
Resultados	**Impacto en la gestión de materias primas**: Mejora en el pronóstico de pedidos. **Mejor modelo**: LSTM mostró el mejor rendimiento en pronósticos. **Datos**: Pronósticos realizados con 135 semanas de datos históricos para 10 semanas futuras. **Limitaciones**: No se consideraron fechas ni eventos especiales, lo que dificultó identificar tendencias.

b) Selección de técnicas de predicción

Para la selección de las técnicas de predicción, es fundamental realizar una comparación exhaustiva de los casos de estudio, ya que esto permite tomar decisiones informadas sobre cuál modelo utilizar. Al analizar las ventajas y desventajas de los modelos empleados en diferentes contextos, se puede seleccionar la técnica más adecuada según las características específicas de cada situación, como el tipo de productos, los patrones de demanda y los objetivos organizacionales. Esta comparación facilita una implementación más efectiva y optimizada de los modelos de predicción, garantizando una gestión de inventarios más precisa y eficiente. Además, al comprender el rendimiento de cada modelo en distintos escenarios, se pueden identificar posibles mejoras o ajustes que optimicen los resultados según las necesidades particulares de la organización, contribuyendo a una mejor toma de decisiones estratégicas en la gestión de inventarios.

Tabla 13. Comparación de modelos utilizados en los distintos casos de estudio.

Título	Modelos Utilizados	Ventajas	Desventajas
Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization	ARIMA-NARNN, XGBoost, modelo de optimización combinatoria	• Mejora en el rendimiento de inventario • Incremento del 1.3% en beneficios anuales.	• Dependencia de la calidad de los datos. Requiere experiencia en modelado combinatorio.
Inventory Management and Cost Reduction of Supply Chain Processes Using A-Based Time-Series Forecasting and ANN Modeling	Redes Neuronales Artificiales (ANN), enfoque de pronóstico basado en series de tiempo	• Precisión de pronóstico de 75-80%. • Optimiza la planificación de recursos.	• Puede ser complejo ajustar y entrenar redes neuronales. • Necesidad de datos extensos para un buen rendimiento.
Optimization of inventory management through computer vision and machine learning technologies	AlexNet, VGG16, ResNet, EfficientNet, MobileNet	• Reducción del 45% en el tiempo de conteo de inventario. • Aumento del 9% en la precisión del inventario.	• Requiere hardware potente para entrenamiento y procesamiento. • Implementación puede ser costosa y técnica.
Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis	Random Forest, Gradient Boosting, LightGBM, XGBoost, CatBoost, LSTM, BiLSTM, ARIMA	• LSTM mostró el mejor rendimiento en pronósticos. • Adecuado para series temporales complejas.	• Los modelos complejos pueden ser difíciles de interpretar. • No considera eventos especiales, limitando la precisión.

Tras analizar los casos de estudio presentados en la **Tabla 13:** Comparación de modelos utilizados en los distintos casos de estudio, se determinó que las mejores opciones para el problema en cuestión son los modelos AutoRegressive Integrated Moving Average (ARIMA) y Extreme Gradient Boosting (XGBoost). Estos modelos destacan por su capacidad de adaptarse a diferentes escenarios y ofrecer resultados significativos en la gestión de inventarios y la previsión de series temporales. ARIMA, con su enfoque estadístico, es ideal para datos lineales y estables, mientras que XGBoost, con su capacidad de aprendizaje basado en gradientes, sobresale en escenarios más complejos gracias a su precisión y robustez. Aunque ambos modelos presentan desventajas como la necesidad de datos de calidad y experiencia en

modelado, sus ventajas, como la mejora en la planificación de recursos y el aumento en beneficios operativos, los posicionan como las alternativas más adecuadas para implementar en este proyecto.

c) Selección de la base de datos

La selección de una base de datos adecuada es esencial para el éxito de cualquier proyecto de desarrollo de software. En la **Tabla 14**, se presenta una comparativa entre tres tipos de bases de datos: SQLite, MySQL y MongoDB. Esta tabla resume características clave, como escalabilidad, instalación, soporte de transacciones, modelo de datos, flexibilidad y costo. Al analizar estas bases de datos, se busca entender cuál se adapta mejor a las necesidades específicas del proyecto, asegurando así un manejo óptimo de los datos y una estructura eficiente para el desarrollo y la implementación.

Tabla 14. Tabla comparativa de bases de datos.

Característica	**SQLite**	**MySql**	**MongoDB**
Escalabilidad	Relacional	Relacional	NoSQL
Instalación	Sin instalación (archivo único)	Requiere servidor	Requiere servidor
Soporte de Transacciones	Sí (ACID)	Sí (ACID)	No (eventual consistency)
Modelo de Datos	Tablas con filas y columnas	Tablas con filas y columnas	Documentos JSON
Flexibilidad	Menos flexible (estructura fija)	Estructura fija	Muy flexible (sin esquema)
Costo	Gratuita y de código abierto	Gratuita (versión comunitaria)	Gratuita (versión comunitaria)

SQLite se presenta como una opción ideal para sistemas web que requieren una solución ligera y de fácil implementación. Su naturaleza elimina la necesidad de un servidor separado, lo que simplifica el despliegue y reduce los costos operativos. Además, a pesar de sus limitaciones en escalabilidad y flexibilidad, ofrece soporte completo para transacciones atomicidad, consistencia, aislamiento y durabilidad **(**ACID), garantizando la integridad de los datos. Por estas razones, SQLite es

particularmente adecuada para aplicaciones de menor escala y entornos de desarrollo, donde la rapidez de implementación y la simplicidad son esenciales[23].

d) Backend

Python es uno de los lenguajes de programación más populares y versátiles, lo que lo convierte en una excelente opción para desarrollar el backend de aplicaciones web. Una de las principales razones para su elección es su simplicidad y legibilidad. La sintaxis clara y concisa de Python facilita el desarrollo y mantenimiento del código, permitiendo a los desarrolladores centrarse en la lógica de negocio en lugar de lidiar con complejidades sintácticas. Esto resulta especialmente beneficioso en entornos colaborativos, donde la claridad del código puede mejorar la comunicación entre los miembros del equipo.

Además, Python cuenta con una amplia comunidad de desarrolladores, lo que significa que hay abundantes recursos, bibliotecas y frameworks disponibles. Herramientas como Flask no solo aceleran el proceso de desarrollo al proporcionar funcionalidades listas para usar, sino que también ofrecen estructuras sólidas para construir aplicaciones escalables y seguras. Esta flexibilidad es crucial para una panadería que podría expandir sus operaciones en el futuro.

Otro aspecto relevante es la capacidad de integración que ofrece Python. Se integra fácilmente con otros lenguajes y tecnologías, lo que permite crear APIs y facilitar la comunicación con bases de datos, servicios externos y otras aplicaciones. Esto es fundamental para garantizar que el sistema informático de la panadería funcione de manera fluida y eficiente[13].

Uso de Python para machine learning

Python se ha consolidado como el lenguaje predilecto para el desarrollo de aplicaciones de machine learning, y las razones son numerosas. En primer lugar, Python ofrece una amplia gama de bibliotecas y frameworks diseñados específicamente para este propósito, como TensorFlow, Keras, scikit-learn y PyTorch. Estas herramientas proporcionan algoritmos y modelos preconstruidos que simplifican

el proceso de implementación, permitiendo a los desarrolladores centrarse en el desarrollo de soluciones en lugar de reinventar la rueda[24].

La facilidad de prototipado es otro factor clave que hace de Python una opción ideal para machine learning. Gracias a su sintaxis intuitiva, los desarrolladores pueden crear prototipos rápidamente, lo que es esencial en este campo, donde la experimentación y la iteración son fundamentales para encontrar el modelo más efectivo. Esto permite a las panaderías realizar análisis predictivos sobre ventas, inventarios y tendencias del mercado de manera más ágil.

Además, Python no solo es fuerte en machine learning, sino que también es excelente para el análisis de datos. Bibliotecas como Pandas y NumPy facilitan la manipulación y análisis de grandes volúmenes de datos, lo que es fundamental para entrenar modelos de machine learning. La combinación de estas capacidades permite a las panaderías optimizar sus operaciones y tomar decisiones informadas basadas en datos[24].

e) Frontend

La elección de la tecnología adecuada para el desarrollo frontend es crucial para la creación de aplicaciones web efectivas y eficientes, ya que impacta directamente en la experiencia del usuario y en la escalabilidad del proyecto. En la **Tabla 15**, se presenta una comparativa entre tres de las principales tecnologías frontend: React, Angular y Vue.js. Esta tabla destaca características clave como el enfoque, rendimiento, ecosistema, comunidad, reutilización de componentes e integración, que son factores determinantes al momento de seleccionar la mejor opción para un proyecto específico. React, Angular y Vue.js son conocidas por sus distintas filosofías y ventajas en el desarrollo de interfaces de usuario, cada una con sus fortalezas y debilidades en términos de facilidad de uso, flexibilidad y compatibilidad con otras tecnologías. Al evaluar estas tecnologías, se busca identificar cuál se adapta mejor a los requisitos del proyecto, asegurando no solo una experiencia de usuario óptima, sino también un desarrollo ágil y un mantenimiento más sencillo a largo plazo. Además, una elección adecuada de la tecnología frontend contribuye a mejorar la eficiencia del equipo de desarrollo, optimizando tiempos y recursos durante las fases de implementación y escalabilidad de la aplicación.

Tabla 15. Cuadro comparativo de tecnologías frontend.

Característica	React	Angular	Vue.js
Enfoque	Biblioteca centrada en la creación de interfaces de usuario	Framework completo para el desarrollo de aplicaciones	Framework progresivo, adaptable a necesidades
Rendimiento	Alto rendimiento gracias al Virtual DOM	Rendimiento sólido, pero más pesado debido a su tamaño	Muy buen rendimiento y bajo tamaño
Ecosistema	Amplio ecosistema de bibliotecas y herramientas	Ecosistema robusto, pero más pesado	Ecosistema creciente y flexible
Comunidad	Comunidad muy activa y recursos abundantes	Comunidad activa y soporte sólido	Comunidad en crecimiento, buena documentación
Reutilización de Componentes	Alta reutilización de componentes gracias a su arquitectura	Buena reutilización, pero menos flexible	Alta reutilización de componentes
Integración	Se integra fácilmente con otras bibliotecas y frameworks	Integración más compleja debido a su estructura	Fácil integración con otras bibliotecas

React se destaca en varios aspectos clave, como su facilidad de aprendizaje y su enfoque en la creación de interfaces de usuario, lo que lo hace ideal para proyectos que requieren flexibilidad y rendimiento. Su alto rendimiento gracias a la capacidad de reutilizar componentes hace que sea una excelente opción para aplicaciones interactivas y dinámicas. Además, su comunidad activa y el amplio ecosistema de bibliotecas disponibles brindan un soporte robusto para los desarrolladores[25].

3.1.4 Definición de la metodología

En el contexto del desarrollo del proyecto web, es fundamental seleccionar una metodología ágil adecuada que se ajuste a las características y necesidades específicas del mismo. A continuación, en la **Tabla 16**, se presenta una comparativa entre tres metodologías ágiles ampliamente reconocidas: Extreme Programming (XP), Kanban

y Scrum. Esta tabla tiene como objetivo proporcionar una comprensión más detallada de las características de cada metodología y su aplicación práctica en proyectos de desarrollo. Al analizar estas metodologías, podré identificar cuál se adapta mejor a la dinámica de mi proyecto y optimizar así el proceso de desarrollo, asegurando una mejor calidad en el resultado final. Además, este enfoque permitirá establecer un marco de trabajo eficiente que fomente la colaboración y la adaptación al cambio, maximizando los recursos disponibles y los tiempos de entrega.

Tabla 16. Tabla comparativa de metodologías agiles.

Característica	**Extreme Programming (XP)**	**Kanban**	**Scrum**
Tamaño del proyecto	Pequeños y medianos.	Pequeños, medianos y grandes.	Proyectos de cualquier tamaño y complejidad
Roles	Desarrolladores, clientes, coach.	No hay roles definidos; todos participan en el flujo.	Scrum Master, Product Owner, Equipo de Desarrollo.
Tablero	Uso de tablas para seguimiento de tareas.	Tablero Kanban para visualizar el flujo de trabajo.	Tablero Scrum para seguimiento de tareas en Sprints.
Énfasis en pruebas	desarrollo dirigido por pruebas.	Enfoque en la mejora del flujo más que en pruebas.	Revisión del trabajo al final de cada Sprint.
Ventajas	Alta calidad de código, satisfacción del cliente.	Flexibilidad y adaptabilidad al flujo de trabajo.	Estructura clara, roles definidos, foco en el cliente.
Desventajas	Puede ser complejo de implementar, requiere disciplina.	Puede resultar en falta de dirección si no se gestiona bien.	Rigidez en las iteraciones, puede ser difícil adaptarse a cambios.

Después de analizar detenidamente la información sobre las metodologías, se tomó la decisión de optar por XP como la metodología más adecuada para el proyecto en cuestión. Esta elección se basó en las características distintivas de XP, que enfatizan la calidad del código y la colaboración constante con el cliente. La metodología XP permite adaptarse a cambios frecuentes en los requisitos, lo que resulta fundamental en entornos dinámicos. Además, sus prácticas, como la programación en pareja y las

pruebas automatizadas, garantizan un producto final de alta calidad, alineado con las expectativas del cliente. La capacidad de XP para fomentar un ambiente de trabajo colaborativo y su enfoque en la mejora continua la convierten en la opción ideal para maximizar el éxito del desarrollo del proyecto. Esta decisión también asegura que el equipo pueda responder eficientemente a los desafíos que surjan durante las etapas críticas del desarrollo[26].

3.2 Desarrollo de la propuesta

La implementación de la metodología Extreme Programming (XP) en el desarrollo de mi sistema informático para la gestión de productos en la panadería y pastelería "Flor de Cebada" se llevará a cabo en varias etapas que reflejan los principios fundamentales de XP. A continuación, se describen las fases del desarrollo, los roles del equipo, las prácticas clave y los beneficios esperados de aplicar XP en este proyecto.

3.2.1 Fase I: Planificación

En esta fase, el equipo de desarrollo y los gerentes acuerdan el orden de la implantación de las historias de usuario para poder hacer la entrega de la misma.

a) Requisitos funcionales del sistema

La definición de requisitos es un paso fundamental en el desarrollo del sistema informático para la panadería y pastelería "Flor de Cebada", ya que establece las características y funcionalidades que se cumplió con el software para satisfacer las necesidades operativas y comerciales del negocio. Este sistema está diseñado para facilitar la gestión del inventario, optimizar las ventas y mejorar la eficiencia operativa mediante el uso de tecnologías avanzadas, como modelos de series de tiempo y algoritmos de Machine Learning. A continuación, se detallan los requisitos que guiarán el desarrollo del programa, organizados en diferentes módulos que abarcan desde la interfaz de usuario hasta la gestión de datos y seguridad.

Interfaz de usuario amigable

Dashboard principal: Un panel central donde los empleados puedan ver el estado actual del inventario, las ventas recientes.

Control de stock: Alertas automáticas cuando los niveles de inventario estén bajos y controle la caducidad de los productos.

Módulo de registro de productos: Una sección para registrar nuevos productos, ingresar información como precios, categorías, y fechas de expiración.

Módulo de ventas: Interfaz para registrar ventas diarias, integrando el escáner de códigos para agilizar el proceso de facturación.

Gestión de inventario con series de tiempo

Predicción de demanda: Implementar modelos de series de tiempo para predecir la demanda futura de los productos, lo que permitirá ajustar la producción diaria o semanal.

Detección de tendencias: Seguimiento detallado de las ventas a lo largo del tiempo, permitiendo analizar patrones de consumo.

Análisis de ventas

Reportes Automatizados: Generación de reportes diarios, semanales, y mensuales sobre las ventas.

b) Roles Asignados en el proyecto

En la metodología XP (Extreme Programming), se distribuyen distintos roles entre los integrantes del equipo con el objetivo de fomentar la colaboración y la comunicación constante. Cada rol está diseñado para cubrir necesidades específicas dentro del proceso de desarrollo, promoviendo así un ambiente de trabajo más dinámico y eficiente. Para este proyecto, se seleccionaron únicamente aquellos roles que se ajustan a sus necesidades específicas, permitiendo una adaptación óptima del equipo a las particularidades de este. Esto asegura que los miembros del equipo puedan concentrarse en sus fortalezas y aportar de manera más efectiva al éxito del proyecto.

Tabla 17. Cuadro de definición de roles.

Nombre	Rol	Función
Jean Carlo Freire	Programador	Desarrollar el código fuente del sistema y realizar pruebas iniciales
Ing. Clay Aldas	Entrenador-Rastreador	Monitorea las actividades y el progreso del proyecto, ofreciendo soluciones y recomendaciones fundamentadas en su experiencia.
Silvana Valencia	Cliente-Tester	Proporcionar requisitos, validar las funcionalidades y realizar pruebas finales.

c) Arquitectura de la aplicación

La estructura de la aplicación estará diseñada utilizando diferentes herramientas. En el backend se implementará Python junto con el framework Flask, mientras que en el frontend se optará por React para garantizar una interfaz interactiva y funcional. La base de datos estará gestionada con SQLite.

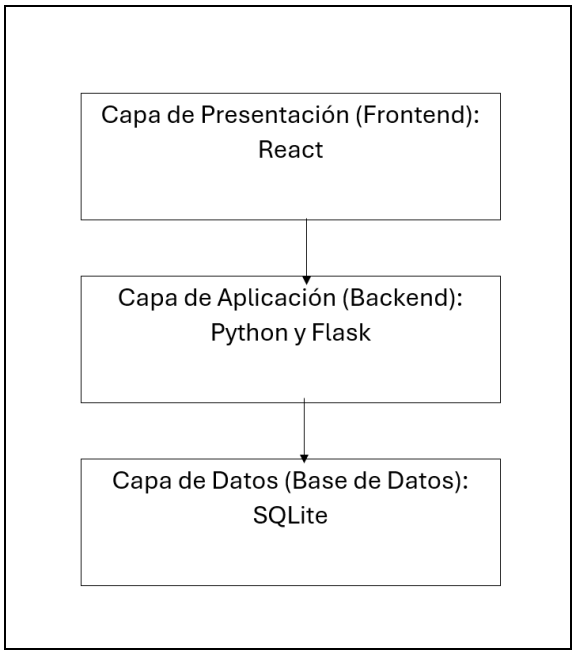


Figura 13. Capas de arquitectura de la aplicación.

d) Requerimientos de Software

Las herramientas empleadas para la creación del sistema son las siguientes:

Visual Studio Code: Es un entorno de desarrollo integrado ligero y versátil que ofrece soporte para múltiples lenguajes de programación, incluyendo Python y JavaScript. Facilita la edición de código, depuración y la integración con control de versiones.

React: Es una biblioteca de JavaScript utilizada para construir interfaces de usuario dinámicas y reactivas. Permite desarrollar componentes reutilizables que optimizan la experiencia del usuario en aplicaciones web.

Flask: Es un framework ligero de Python para el desarrollo de aplicaciones web. Proporciona una estructura sencilla para crear y manejar la lógica del backend de manera eficiente.

SQLite: Es un sistema de gestión de bases de datos ligero y embebido, ideal para aplicaciones pequeñas y medianas. Permite almacenar datos estructurados sin necesidad de un servidor externo.

CSS: Hojas de estilo en cascada que controlan el diseño y la apariencia visual de los elementos HTML en el frontend, mejorando la presentación y experiencia de usuario.

JavaScript: Lenguaje de programación que potencia la interactividad del frontend, permitiendo que React maneje eventos y actualice dinámicamente los elementos en pantalla.

HTML: Lenguaje de marcado estándar para la creación de la estructura y contenido de las páginas web del sistema.

e) Historias de Usuario

El presente proyecto utiliza historias de usuario como herramienta clave para garantizar que cada requisito sea abordado de forma clara, concisa y priorizada, facilitando una gestión efectiva del alcance del proyecto. Este enfoque permite asegurar una alineación constante con las metas del negocio, así como con las necesidades y expectativas del usuario final. Las historias de usuario son

fundamentales para mantener un enfoque centrado en el cliente, ya que proporcionan un marco que facilita la comprensión de lo que realmente importa al usuario y a la organización.

Tabla 18. Modelo de historias de usuario.

Historia de Usuario	
Número:	**Usuario:**
Nombre Historia:	
Prioridad Negocio:	**Riesgo en Desarrollo:**
Puntos Estimados:	**Iteración Asignada:**
Programador Responsable:	
Descripción:	
Observación:	

Número y usuario: Identificación única de la historia y el tipo de usuario asociado.

Nombre de la historia: Resumen breve y descriptivo del objetivo principal.

Prioridad de negocio y riesgo en desarrollo: Parámetros que ayudan a determinar el orden de implementación y los posibles desafíos asociados.

Puntos estimados e iteración asignada: Indicadores del esfuerzo necesario y el momento en el cual será implementada.

Programador responsable: Miembro del equipo asignado a la historia.

Descripción: Detalle sobre las funcionalidades esperadas y el propósito de la historia.

Observaciones: Notas adicionales o cambios realizados durante el proceso.

Para la elaboración del proyecto se desarrollaron las siguientes historias:

Tabla 19. Historia de Control de acceso del sistema.

Historia de Usuario	
Número: 1	**Usuario:** Administrador
Nombre Historia: Control de acceso al sistema	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 3	**Iteración Asignada:**1
Programador Responsable: Jean Carlo Freire	
Descripción: El administrador puede ingresar al sistema mediante un inicio de sesión con mi usuario y contraseña autorizadas. El sistema debe validar credenciales.	
Observación: ninguna	

Tabla 20. Historia Registro de nuevos Usuarios.

Historia de Usuario	
Número: 2	**Usuario:** Administrador
Nombre Historia: Registro de nuevos usuarios en el sistema	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 4	**Iteración Asignada:**1
Programador Responsable: Jean Carlo Freire	
Descripción: el responsable puede registrar nuevos usuarios en el sistema proporcionando información como nombre, correo electrónico y contraseña inicial.	
Observación: El sistema debe validar los datos ingresados y verificar que el correo no esté registrado previamente.	

Tabla 21. Historia de diseño de dashboard principal.

Historia de Usuario	
Número: 3	**Usuario:** Administrador
Nombre Historia: Diseño del Dashboard Principal	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 5	**Iteración Asignada:**1
Programador Responsable: Jean Carlo Freire	
Descripción: El usuario, al iniciar sesión, accede a un dashboard principal que muestra el logo de la empresa en una posición destacada. El dashboard incluye botones de navegación para ingresar a los diferentes apartados del sistema, como: Inicio, Gestión de Productos, Ventas, Reportes, Reporte de Ventas. El dashboard también cuenta con un botón para cerrar sesión de manera segura. El diseño es intuitivo, responsivo y adaptado a dispositivos de diferentes tamaños.	
Observación: El diseño reflejar la identidad corporativa de la empresa y garantizar un acceso rápido a las funcionalidades más utilizadas.	

Tabla 22. Historia de Control de productos de caducidad y de stock.

Historia de Usuario	
Número: 4	**Usuario:** Administrador
Nombre Historia: Visualización de productos con bajo stock y próximos a caducar	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 4	**Iteración Asignada:**2
Programador Responsable: Jean Carlo Freire	
Descripción: Al iniciar sesión, el administrador ve en la pantalla principal una lista destacada de productos que tienen bajo stock y aquellos que están próximos a caducar. • La sección de **Bajo Stock** debe mostrar el nombre del producto, la cantidad actual. • La sección de **Próximos a Caducar** debe incluir el nombre del producto, la fecha de caducidad y los días restantes.	
Observación: El administrador debe recibe alertas visuales para los productos.	

Tabla 23. Historia de gestión de inventario.

Historia de Usuario	
Número: 5	**Usuario:** Administrador
Nombre Historia: Gestión de inventario	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 5	**Iteración Asignada:** 2
Programador Responsable: Jean Carlo Freire	
Descripción: El administrador puede registrar, modificar, consultar y eliminar productos del inventario para tener un control actualizado de los productos disponibles.	
Observación:	

Tabla 24. Historia de gestión de ventas.

Historia de Usuario	
Número: 6	**Usuario:** Vendedor
Nombre Historia: Gestión de ventas	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 4	**Iteración Asignada:** 2
Programador Responsable: Jean Carlo Freire	
Descripción: El vendedor registra las ventas de los productos utilizando un escáner de códigos.	
Observación: El sistema debe calcula automáticamente subtotales y totales.	

Tabla 25. Historia de generación de nota en pdf.

Historia de Usuario	
Número: 7	**Usuario:** Vendedor
Nombre Historia: Generación de nota de venta en PDF	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 3	**Iteración Asignada:** 2
Programador Responsable: Jean Carlo Freire	
Descripción: Al finalizar una venta, el sistema genera automáticamente una nota de venta con los detalles del cliente, productos vendidos, cantidades, subtotales, impuestos y el total. La nota de venta estará disponible para descargar en formato PDF.	
Observación: El diseño del PDF debe incluir el logo de la empresa y cumplir con los requisitos legales de facturación.	

Tabla 26. Historia de generación de reportes.

Historia de Usuario	
Número: 8	**Usuario:** Administrador
Nombre Historia: Generación de reportes	
Prioridad Negocio: Alta	**Riesgo en Desarrollo**: Medio
Puntos Estimados: 4	**Iteración Asignada:** 3
Programador Responsable: Jean Carlo Freire	
Descripción: Se generan reportes detallados de ventas semanales y mensuales para analizar el desempeño y tomar decisiones informadas.	
Observación:	

Tabla 27. Historia de control de acceso del sistema.

Historia de Usuario	
Número: 9	**Usuario:** Administrador
Nombre Historia: Generación de predicciones	
Prioridad Negocio: Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 6	**Iteración Asignada:** 3
Programador Responsable: Jean Carlo Freire	
Descripción: utilizar predicciones basadas en datos históricos para anticipar la demanda y optimizar la producción de productos.	
Observación: El sistema debe mostrar las predicciones en un gráfico de tendencias fácil de interpretar.	

Tabla 28. Historia de visualización del historial de ventas.

Historia de Usuario	
Número: 10	**Usuario:** Administrador
Nombre Historia: Visualización del historial de ventas	
Prioridad Negocio: Media	**Riesgo en Desarrollo:** Bajo
Puntos Estimados: 3	**Iteración Asignada:**3
Programador Responsable: Jean Carlo Freire	
Descripción: El administrador puede visualizar un gráfico interactivo que muestra la tendencia de ventas a lo largo del tiempo. El gráfico debe incluir datos históricos, como la cantidad de ventas realizadas	
Observación: El gráfico debe ser fácil de interpretar.	

f) Valoración / Estimación de historias de usuario

Una vez definidos los requisitos funcionales, se procedió a calcular el tiempo estimado para su desarrollo, teniendo en cuenta la complejidad de cada funcionalidad y los recursos disponibles. Este proceso de estimación es crucial para una correcta planificación y gestión del proyecto, ya que permite identificar posibles cuellos de botella, asignar los recursos adecuados y establecer plazos realistas para la entrega de cada parte del sistema. A continuación, se presenta la distribución del tiempo de ejecución para cada requisito funcional, considerando la siguiente escala: esta escala no solo facilita la priorización de tareas, sino que también permite al equipo realizar un seguimiento eficiente del avance del proyecto, asegurando que se cumplan los plazos establecidos sin comprometer la calidad del producto final.

- 1 semana de trabajo comprende 5 días laborales.
- 1 día equivale a 5 horas de trabajo.

Tabla 29. Valoración de historias de usuario.

N.º	Historia de Usuario	Puntos Estimados	Horas	Días	Semanas
1	Control de Acceso	3	15	3	0.6
2	Registro de nuevos usuarios en el sistema	4	20	4	0.8
3	Diseño del Dashboard Principal	5	25	5	1.0
4	Visualización de productos con bajo stock y próximos a caducar	4	20	4	0.8
5	Gestión de Inventario	5	25	5	1.0
6	Gestión de Ventas	4	20	4	0.8
7	Generación de nota de venta en PDF	3	15	3	0.6
8	Generación de Reportes	4	20	4	0.8
9	Predicciones de la demanda	6	30	6	1.2
10	Visualización del historial de ventas	3	15	3	0.6
Total		**25**	**205**	**25**	**8.2**

g) Plan de entrega

Después de estimar las historias de usuario, se elabora un plan de entrega para el desarrollo del proyecto, especificando la iteración a la que corresponde cada una. Este plan incluye plazos definidos para cada iteración, asegurando un seguimiento claro del progreso y permitiendo realizar ajustes según sea necesario

Tabla 30. Plan de entrega.

N.º	Historia de Usuario	Horas	Días	Iteraciones		
				1	**2**	**3**
1	Control de Acceso	15	3	X		
2	Registro de nuevos usuarios	20	4	X		
3	Diseño del Dashboard Principal	25	5	X		
4	Visualización de productos con bajo stock y próximos a caducar	20	4		X	
5	Gestión de Inventario	25	5		X	
6	Gestión de Ventas	20	4		X	
7	Generación de nota de venta en PDF	15	3		X	
8	Generación de Reportes	20	4			X
9	Predicción de Demanda	30	6			X
10	Visualización del historial de ventas	15	3			X

3.2.2. Fase II: Diseño

En esta etapa se desarrollan las iteraciones, que implican realizar avances graduales y manejables dentro de un periodo de tiempo breve.

a) Iteracion 1

Conforme a la tabla el plan de entrega se distribuye en 3 iteraciones en la primera iteración se hacen las actividades de Control de acceso, Registro de Usuarios y el Diseño del Dashboad principal.

Tabla 31. Historia de control de iteración 1.

Código	Historia de Usuario	Prioridad	Riesgo
1	Control de acceso	Alta	Alto
2	Registro de nuevos usuarios	Alto	Medio
3	Diseño del Dashboard Principal	Alto	Medio

Ingreso al sistema

Permite acceder al sistema utilizando el nombre de usuario y la contraseña.

Tabla 32. Actividades de ingreso del sistema.

Historia N: 1	**Responsable:** Jean Carlo Freire
Actividades	
• Verificación de credenciales de usuario en la base de datos • Desarrollo de la interfaz de usuario con el logo de la empresa	

Registro de nuevos usuarios

Permite crear nuevos usuarios con información personal de los mismos.

Tabla 33. Actividades de registro de nuevos usuarios.

Historia N: 2	**Responsable:** Jean Carlo Freire
Actividades	
• Verificación de registro de datos personales en la base de datos . • Desarrollo de la interfaz de usuario para el registro de un nuevo usuario.	

Diseño del dashboard principal

Desarrollo de la página principal con todos los apartados necesarios para poder navegar entre páginas y además el cierre de sesión.

Tabla 34. Actividades de diseño del dashboard principal.

Historia N: 3	**Responsable:** Jean Carlo Freire
Actividades	
• Generación de los componentes necesarios para la navegación entre páginas en la frontend • Generación de botones y el cierre de sesión	

b) Iteración 2

En la Iteración 2 se realiza Visualización de productos con bajo stock y próximos a caducar, Gestión de inventario, Gestión de ventas, Generación de nota de venta en PDF.

Tabla 35. Historia de control de iteración 2.

Código	**Historia de Usuario**	**Prioridad**	**Riesgo**
4	Visualización de productos con bajo stock y próximos a caducar	Alta	Medio
5	Gestión de inventario	Alto	Alto
6	Gestión de ventas	Alto	Medio
7	Generación de nota de venta en PDF	Alto	Medio

Visualización de productos con bajo stock y próximos a caducar

En la página principal de inicio se puede visualizar información sobre la empresa además de mostrar los productos que tienen bajo stock y los que están próximos a caducar.

Tabla 36. Actividades de visualización de productos con bajo stock y próximos a caducar.

Historia N: 4	**Responsable:** Jean Carlo Freire
Actividades	
• Consultar la base de datos para verificar el stock y las fechas de caducidad. • Generar alertas cuando el stock de un producto sea de cinco unidades o menos. • Emitir alertas tres días antes de la fecha de caducidad de un producto.	

Gestión de inventario

El Usuario puede añadir, editar y eliminar productos estos tienen filtros para poder buscar más rápido.

Tabla 37. Actividades de gestión de inventario.

Historia N: 5	**Responsable:** Jean Carlo Freire
Actividades	
• Configurar el módulo de gestión de inventario para añadir, editar y eliminar productos. • Implementar filtros avanzados para búsqueda rápida por nombre, categoría, fecha de caducidad o stock. • Verificar la correcta actualización de datos al añadir, editar o eliminar productos en la base de datos.	

Gestión de ventas

El sistema permite agregar productos a la venta utilizando un lector de código de barras. Durante el proceso, es posible editar la cantidad de productos necesarios o eliminar un producto si se requiere.

Tabla 38. Actividades de gestión de ventas.

Historia N: 6	**Responsable:** Jean Carlo Freire
Actividades	
• Integrar el lector de código de barras con el sistema de gestión de ventas. • Implementar la funcionalidad para agregar productos automáticamente mediante el escaneo del código de barras. • Configurar opciones para editar la cantidad de productos seleccionados durante el proceso de venta. • Incorporar la funcionalidad para eliminar productos seleccionados de la lista de venta en caso necesario.	

Generación de nota de venta en PDF

Tras completar una venta, el sistema permite descargar una nota de venta en formato PDF. Esta nota incluye el logo de la empresa, un detalle de los productos vendidos, los subtotales y el total de la transacción.

Tabla 39. Actividades de generación de nota de venta en PDF.

Historia N: 7	**Responsable:** Jean Carlo Freire
Actividades	
• Diseñar la plantilla de la nota de venta en formato PDF, incluyendo el logo de la empresa y el formato adecuado. • Configurar el sistema para generar automáticamente la nota de venta tras la finalización de cada transacción. • Incluir en la nota de venta un listado detallado de los productos vendidos, sus precios unitarios, subtotales y el total. • Implementar la funcionalidad para descargar la nota de venta en PDF directamente desde la interfaz del sistema.	

c) Iteración 3

En esta tercera iteración conforma la Generación de reportes, Predicción de la demanda y Visualización del historial de Ventas.

Tabla 40. Historia de control de iteración 3.

Código	Historia de Usuario	Prioridad	Riesgo
8	Generación de reportes	Alta	Medio
9	Predicción de demanda	Alta	Alto
10	Visualización del historial de ventas	Media	Bajo

Generación de reportes

Se genera reportes de los productos más vendidos cada día, semana y mes.

Tabla 41. Actividades de generación de reportes.

Historia N: 8	**Responsable:** Jean Carlo Freire
Actividades	
• Generación de consulta a la base datos para filtrar los productos por día, mes y año. • Generación de pdf con gráficos y resultado de la consulta.	

Predicción de demanda

Mediante el código del producto se puede realizar predicción de la demanda en de los próximos 7 y 30 días además de un gráfico que representa esto, así como un gráfico de identificación de tendencias.

Tabla 42. Actividades de predicción de la demanda.

Historia N: 9	**Responsable:** Jean Carlo Freire
Actividades	
• Recolectar y preparar datos históricos de demanda de productos. • Desarrollar el modelo de predicción para calcular la demanda de los próximos 7 y 30 días. • Implementar gráficos de predicción y tendencias en el sistema.	

Visualización del historial de ventas

Esta funcionalidad permite a los usuarios consultar el historial de ventas de los productos registrados en el sistema. Se incluirá una interfaz gráfica que mostrará detalles como fechas, cantidades vendidas y totales.

Tabla 43. Visualización del historial de ventas.

Historia N: 10	**Responsable:** Jean Carlo Freire
Actividades	
• Recolectar y organizar los datos históricos de ventas almacenados en el sistema. • Diseñar e implementar una interfaz gráfica para mostrar el historial de ventas de forma clara y ordenada. • Incorporar filtros y opciones de búsqueda para facilitar el análisis por producto, fecha y rango de tiempo.	

d) Tarjetas CRC

Una tarjeta CRC (Clase, Responsabilidades, Colaboradores) es una herramienta utilizada en el desarrollo de software para representar de manera clara y concisa las funciones de una clase, sus responsabilidades y los colaboradores con los que interactúa. Este enfoque permite a los desarrolladores visualizar de forma sencilla cómo cada clase debe comportarse dentro del sistema y cómo se relaciona con otras clases. Facilita la comprensión de la estructura del sistema y las interacciones entre los diferentes componentes, lo que mejora la organización y la colaboración durante el proceso de desarrollo. Además, las tarjetas CRC fomentan la reflexión y el análisis del diseño del sistema, permitiendo identificar posibles mejoras o áreas de optimización antes de implementar el código, contribuyendo así a un desarrollo más eficiente y coherente. Al ser una herramienta visual y colaborativa, promueve la comunicación entre los miembros del equipo, lo que resulta en una mayor alineación de objetivos y un diseño más sólido y comprensible. Esto, a su vez, mejora la calidad del producto final, reduciendo el riesgo de errores y facilitando el mantenimiento del software a largo plazo.

Tabla 44. Tarjeta CRC de ingreso al sistema.

Ingreso al Sistema	
Responsabilidad	**Colaboradores**
• Verificar credenciales de usuario en la base de datos. • Mostrar la interfaz de usuario con el logo de la empresa.	• Base de datos de usuarios (para la verificación de credenciales). • Módulo de interfaz gráfica (para mostrar la interfaz con el logo).
Observaciones:	

Tabla 45. Tarjeta CRC de registro de nuevos usuarios.

Registro de Nuevos Usuarios	
Responsabilidad	**Colaboradores**
• Verificar el registro de datos personales en la base de datos. • Mostrar la interfaz de usuario para el registro de nuevos usuarios.	• Base de datos de usuarios (para almacenar y validar los datos personales). • Módulo de interfaz gráfica (para mostrar la pantalla de registro).
Observaciones: Es importante implementar validaciones adecuadas en los campos de entrada para garantizar la integridad de los datos personales ingresados por los usuarios.	

Tabla 46. Tarjeta CRC de dashboard principal.

Dashboard Principal	
Responsabilidad	**Colaboradores**
• Generar los componentes necesarios para la navegación entre páginas en el frontend. • Implementar los botones de interacción y la funcionalidad de cierre de sesión.	• Módulo de navegación (para gestionar las transiciones entre páginas). • Módulo de autenticación (para implementar el cierre de sesión).
Observaciones: Es recomendable asegurar una navegación intuitiva y fluida para el usuario, además de proteger el flujo de cierre de sesión para evitar vulnerabilidades de seguridad.	

Tabla 47. Tarjeta CRC de visualización de productos con bajo stock y próximos a caducar.

Visualización de Productos con Bajo Stock y Próximos a Caducar	
Responsabilidad	**Colaboradores**
• Consultar la base de datos para verificar el stock y las fechas de caducidad de los productos. • Generar alertas cuando el stock de un producto sea de cinco unidades o menos. • Emitir alertas tres días antes de la fecha de caducidad de un producto.	• Base de datos de inventario (para obtener el stock y las fechas de caducidad). • Módulo de alertas (para mostrar notificaciones de bajo stock y caducidad). • Dashboard principal (para visualizar la información de alertas).
Observaciones:	

Tabla 48. Tarjeta CRC de gestión de inventario.

Gestión de Inventario	
Responsabilidad	**Colaboradores**
• Configurar el módulo de gestión para añadir, editar y eliminar productos. • Implementar filtros avanzados para facilitar la búsqueda rápida de productos por nombre, categoría, fecha de caducidad o stock. • Verificar que los datos se actualicen correctamente en la base de datos tras cualquier modificación.	• Base de datos de inventario (para gestionar los datos de los productos). • Módulo de interfaz gráfica (para implementar los filtros y las opciones de gestión).
Observaciones: • Es importante garantizar la sincronización adecuada entre la base de datos y la interfaz para evitar inconsistencias en los datos de los productos. • Se deben realizar pruebas periódicas para verificar que los filtros funcionen correctamente y muestren resultados precisos según los parámetros ingresados.	

Tabla 49. Tarjeta CRC de gestión de ventas.

Gestión de ventas	
Responsabilidad	**Colaboradores**
• Integrar el lector de código de barras con el sistema para facilitar la gestión de ventas. • Agregar productos automáticamente al escanear su código de barras. • Permitir la edición de la cantidad de productos seleccionados durante el proceso de venta. • Incorporar la funcionalidad para eliminar productos de la lista de venta si es necesario.	• Módulo de lector de código de barras (para la captura automática de productos). • Base de datos de inventario (para validar la existencia y detalles de los productos). • Interfaz de ventas (para gestionar la edición y eliminación de productos).
Observaciones: La integración del lector de código de barras debe ser fluida y rápida, para evitar tiempos de espera prolongados durante el proceso de venta.	

Tabla 50. Tarjeta CRC de generación de nota de venta PDF.

Generación de nota de venta en PDF	
Responsabilidad	**Colaboradores**
• Diseñar la plantilla de la nota de venta en formato PDF, incluyendo el logo de la empresa y un formato profesional. • Configurar el sistema para generar automáticamente la nota de venta al finalizar una transacción. • Incluir en el PDF un listado detallado de productos vendidos, precios unitarios, subtotales y el total de la transacción. • Implementar la funcionalidad para descargar la nota de venta en PDF desde la interfaz del sistema.	• Módulo de gestión de ventas (para extraer los datos de la transacción). • Biblioteca de generación de PDF (para crear y exportar el documento). • Interfaz gráfica (para habilitar la descarga del PDF).
Observaciones: Es recomendable garantizar que el formato PDF sea compatible con diversos dispositivos y que incluya un diseño claro y fácil de leer para el cliente. verificar la correcta generación de datos para evitar errores en los cálculos del total y subtotales.	

Tabla 51. Tarjeta CRC de generación de reportes.

Generación de reportes	
Responsabilidad	**Colaboradores**
• Consultar la base de datos para filtrar los productos más vendidos por día, semana y mes. • Crear un reporte en formato PDF que incluya gráficos y los resultados de la consulta.	• Base de datos de ventas (para extraer los datos de los productos vendidos). • Biblioteca de gráficos (para generar visualizaciones claras y profesionales). • Biblioteca de generación de PDF (para consolidar los resultados en un reporte descargable).
Observaciones: Se recomienda optimizar las consultas a la base de datos para manejar grandes volúmenes de información y garantizar que los gráficos sean intuitivos y representen de forma precisa los datos analizados. Los gráficos deben ser diseñados para ser fácilmente comprensibles, con leyendas claras y una escala adecuada que represente de manera precisa los datos analizados.	

Tabla 52. Tarjeta CRC de predicción de la demanda.

Predicciones de demanda	
Responsabilidad	**Colaboradores**
• Recolectar y preparar datos históricos de demanda de productos para el análisis. • Desarrollar un modelo de predicción que calcule la demanda para los próximos 7 y 30 días. • Implementar gráficos interactivos que muestren las predicciones y las tendencias de demanda.	• Base de datos de ventas (para obtener los datos históricos necesarios). • Módulo de predicción (basado en algoritmos de machine learning o series de tiempo). • Biblioteca de gráficos (para representar las predicciones y tendencias de forma visual).
Observaciones: Es crucial validar el modelo de predicción con datos históricos para garantizar su precisión y ajustar parámetros según sea necesario. Además, los gráficos deben ser claros e interpretables por el usuario final.	

Tabla 53. Tarjeta CRC de historial de ventas.

Visualización del historial de ventas	
Responsabilidad	**Colaboradores**
• Recolectar y organizar los datos históricos de ventas almacenados en el sistema. • Diseñar e implementar una interfaz gráfica que muestre el historial de ventas de forma clara y ordenada.	• Base de datos de ventas (para acceder y organizar los datos históricos). • Módulo de interfaz gráfica (para mostrar la información de ventas de manera visual).
Observaciones:	

e) Diseño de modelo de datos

El diseño de la base de datos fue realizado por el investigador en colaboración con el personal de la panadería "Flor de Cebada". Juntos, se identificaron las necesidades específicas de la panadería en cuanto a la gestión de productos, ventas e inventarios, para asegurar que la base de datos cumpla con los requisitos operativos y facilite el manejo eficiente de la información. Este proceso incluyó la creación de tablas, relaciones y estructuras necesarias para optimizar el flujo de datos dentro del sistema, garantizando así su correcto funcionamiento y facilidad de acceso.

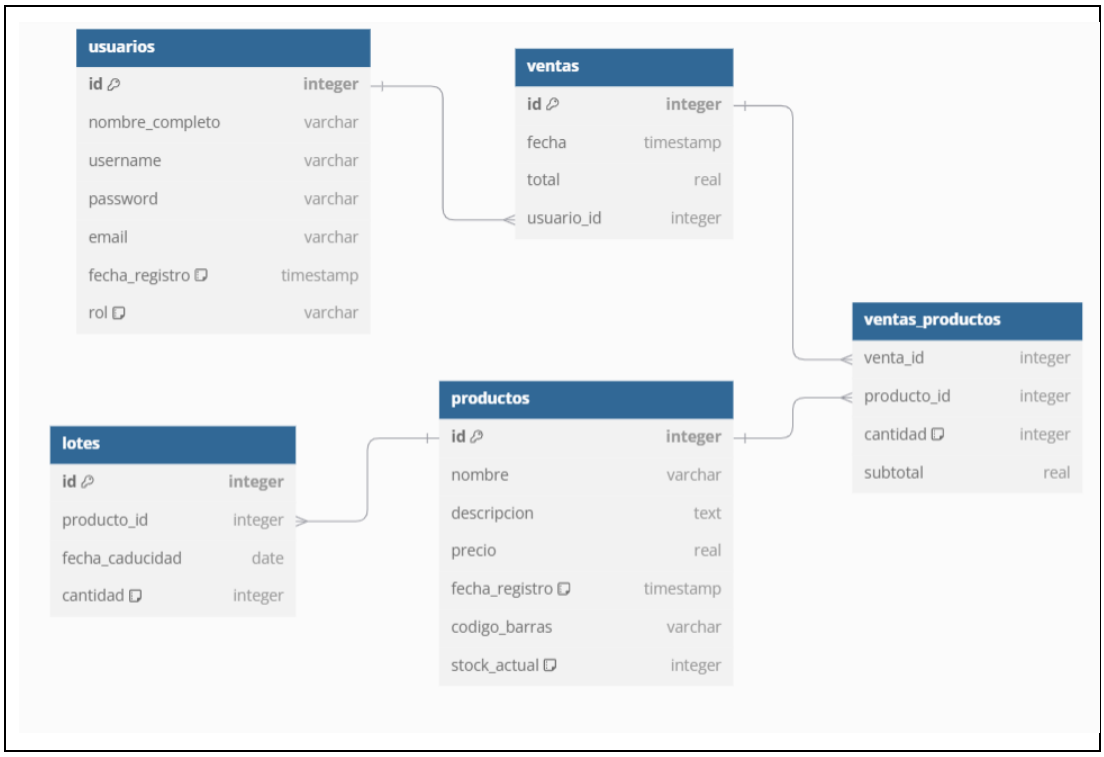


Figura 14. Diagrama de base de datos.

f) Diseño de Iteraciones

A continuación, se presenta un modelado de la aplicación para la cual se utilizó como referencia cada una de las actividades de las distintas Iteraciones.

Iteración 1

Se presenta las figuras pertenecientes a la Interación 1 basándose en el diseño propuesto por el usuario final.

Control de acceso al sistema

Este módulo permite asegurar que solo los usuarios autorizados puedan acceder al sistema mediante un nombre de usuario y una contraseña, protegiendo la información sensible y las operaciones del negocio. Además, se implementan medidas de seguridad adicionales, como la autenticación de dos factores, para fortalecer aún más la protección de los datos y prevenir accesos no autorizados.

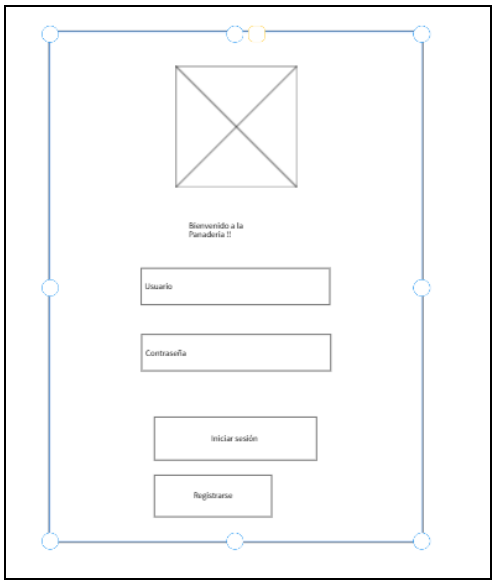


Figura 15. Control de acceso al sistema.

Registro de nuevos usuarios

Permite la creación de cuentas de nuevos usuarios en el sistema, almacenando su información personal para garantizar un acceso adecuado y personalizado según el rol o las necesidades del usuario.

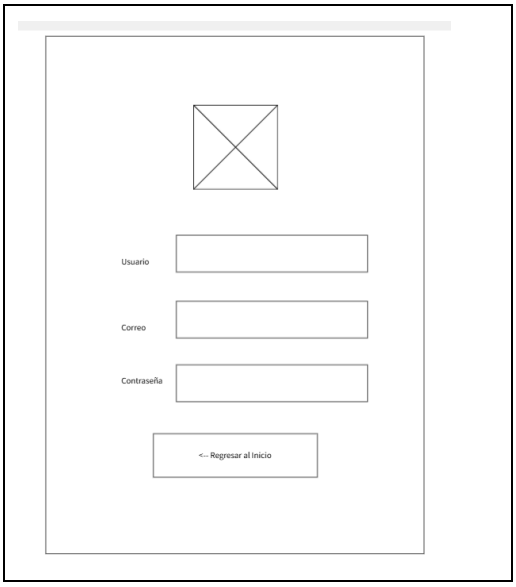


Figura 16. Registro de nuevos usuarios.

Diseño del Dashboard Principal

El dashboard es la página principal del sistema, que incluye una vista general de las operaciones más importantes, como la navegación a otras secciones del sistema y el cierre de sesión. Facilita la interacción con el usuario.

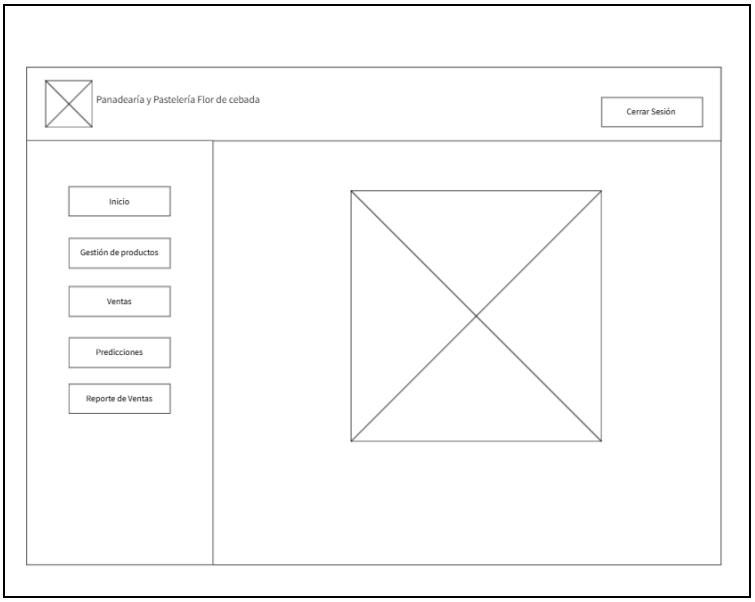


Figura 17. Diseño del dashboard principal.

Iteración 2

Se presenta las figuras pertenecientes a la Interación 2 basándose en el diseño propuesto por el usuario final.

Visualización de productos con bajo stock y próximos a caducar

Muestra alertas visuales sobre los productos que están cerca de agotarse o que están próximos a su fecha de caducidad, ayudando al personal a gestionar el inventario de manera más eficiente. Estas alertas no solo proporcionan información inmediata sobre el estado de los productos, sino que también permiten realizar un seguimiento proactivo para evitar la falta de stock o la pérdida de productos debido a su expiración. Con la integración de colores y notificaciones dinámicas, el sistema facilita la toma de decisiones rápidas y precisas, asegurando que se puedan reponer los productos a tiempo o retirar aquellos que ya no son aptos para la venta, mejorando así la rotación de inventarios y reduciendo las pérdidas económicas.

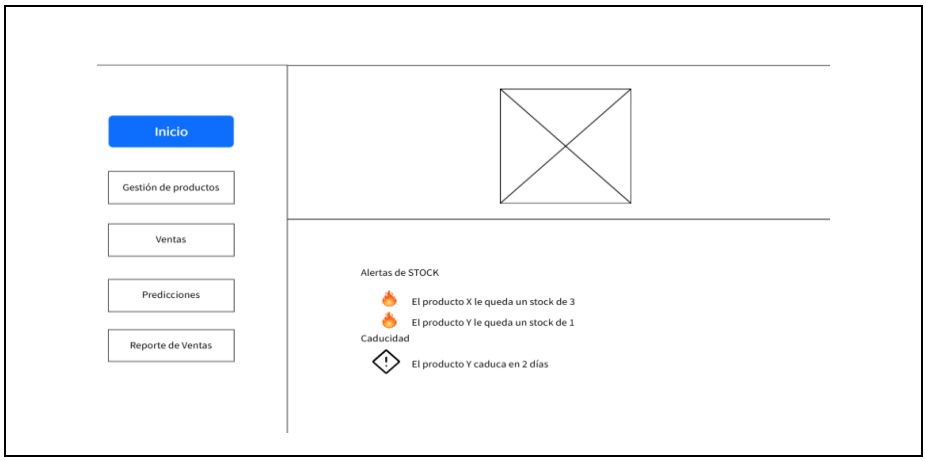


Figura 18. Visualización de productos con bajo stock y próximos a caducar.

Gestión de inventario

Permite añadir, editar y eliminar productos dentro del sistema, además de ofrecer filtros para realizar búsquedas rápidas y fáciles en el inventario, manteniendo un control preciso de los productos disponibles.

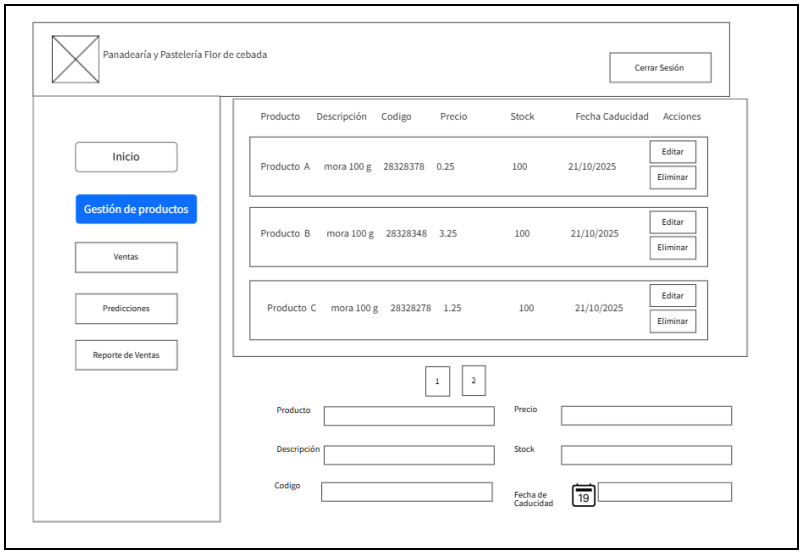


Figura 19. Gestión de inventario.

Gestión de ventas

Este módulo facilita la venta de productos mediante el uso de un lector de códigos de barras, permitiendo agregar productos al carrito, editar cantidades y eliminar productos durante el proceso de venta.

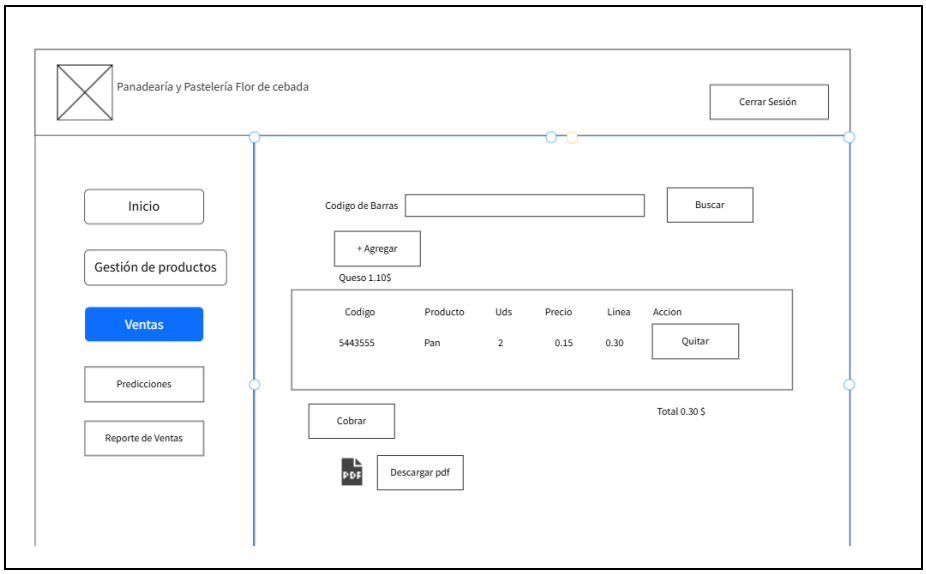


Figura 20. Gestión de ventas.

Generación de nota de venta en PDF

Al finalizar una transacción, el sistema genera una nota de venta en formato PDF, la cual incluye el detalle de los productos vendidos, subtotales, total y el logo de la empresa, y permite la descarga del documento para entrega al cliente, también permite imprimir la nota de venta directamente desde el sistema para una entrega más rápida al cliente.

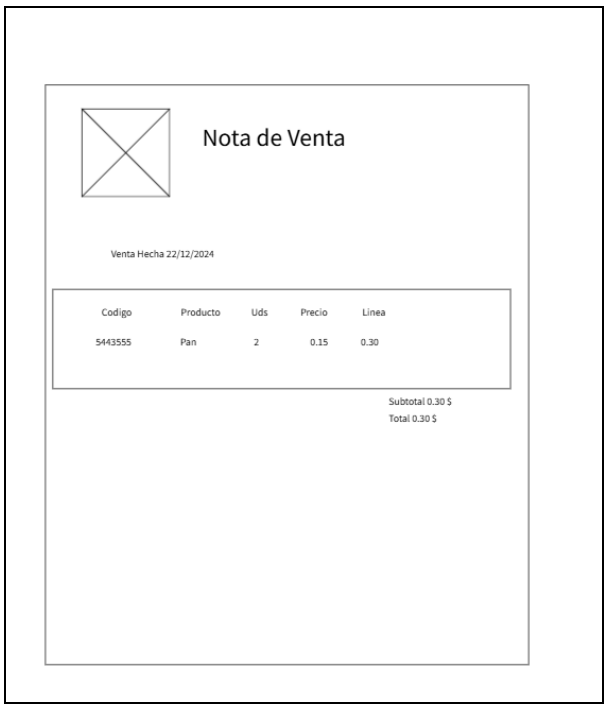


Figura 21. Generación de nota de venta en PDF.

Iteración 3

Se presenta las figuras pertenecientes a la Interación 3 basándose en el diseño propuesto por el usuario final.

Generación de reportes

Permite crear reportes sobre los productos más vendidos en un período específico (día, semana, mes), utilizando gráficos y tablas para facilitar la interpretación y el análisis de los datos.

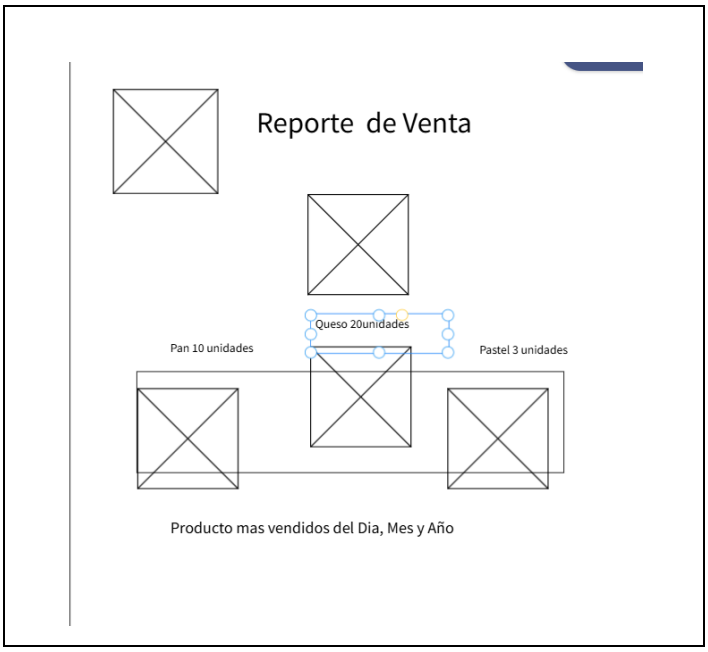


Figura 22. Generación de reportes.

Predicciones de demanda

Utiliza los datos históricos de ventas para prever la demanda de productos en los próximos días (7 y 30 días), permitiendo a la panadería anticiparse a las necesidades de inventario y tomar decisiones informadas.

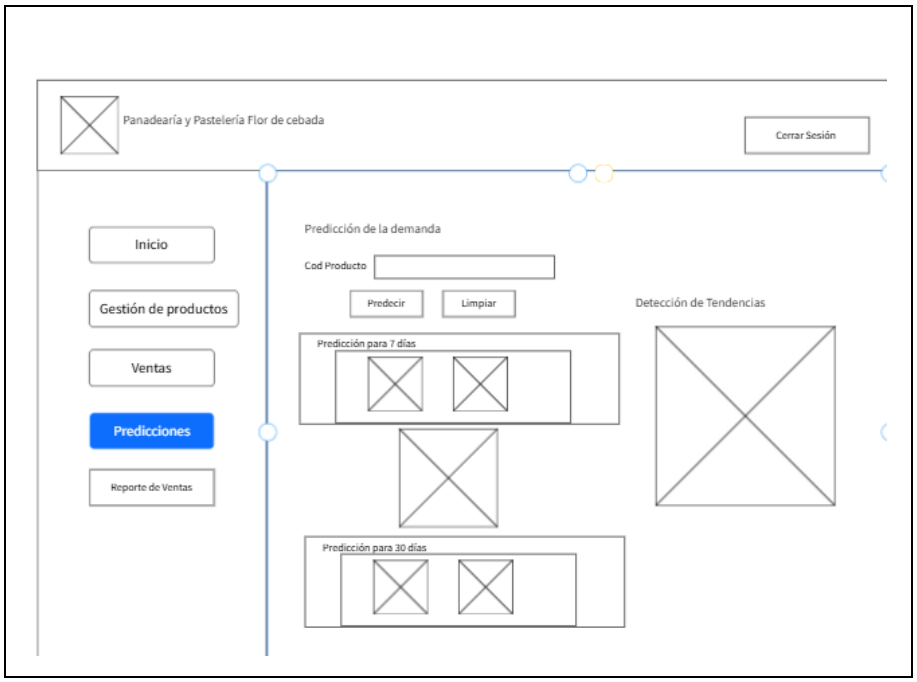


Figura 23. Predicciones de demanda.

Visualización del historial de ventas

Proporciona un registro completo de las ventas realizadas, con detalles como fechas, cantidades vendidas y totales, permitiendo al usuario consultar y analizar las ventas pasadas para evaluar el rendimiento del negocio.



Figura 24. Visualización del historial de ventas.

3.2.3. Fase III: Codificación

a) Creación de la solución en visual Studio Code

Para crear la solución propuesta, se diseñó una arquitectura modular que consta de tres componentes principales: una base de datos, un backend y un frontend. La base de datos y el backend fueron desarrollados en una misma solución, asegurando una integración eficiente y centralizada para la gestión de datos y la lógica del sistema. Por otro lado, el frontend se implementó como una solución independiente, lo que facilita su mantenimiento y escalabilidad. Esta separación permite una interacción clara y organizada entre el backend y el frontend, optimizando el flujo de información y la experiencia del usuario. Además, se aplicaron prácticas de desarrollo ágil para garantizar que cada componente pudiera evolucionar de manera independiente, permitiendo la incorporación de nuevas funcionalidades sin afectar el rendimiento general del sistema. El modularidad también favorece la adaptabilidad, lo que asegura que la solución pueda ajustarse fácilmente a futuros requerimientos o cambios en el entorno de la panadería

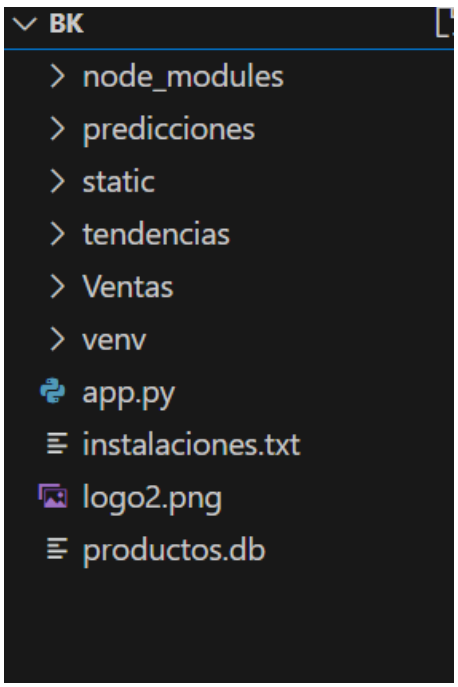


Figura 25. Creación de solución backend.

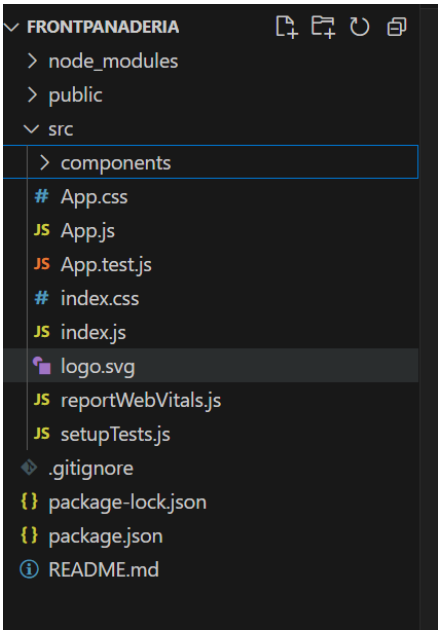


Figura 26. Creación de solución en frontend.

b) Creación y conexión de la base de datos

Se procederá a crear una base de datos estructurada que permita almacenar y gestionar la información necesaria para el sistema. Posteriormente, se conectará al backend para garantizar un flujo de datos eficiente y seguro entre ambos componentes.

```
# Crear tablas si no existen
def create_table():
    # Verificar si la base de datos ya existe o no
    if not os.path.exists(DB_PATH):
        print("Base de datos no encontrada, creando nueva base de datos...")
    else:
        print("Base de datos encontrada, creando tablas si no existen...")

    with get_db_connection() as conn:
        # Crear tabla productos
        conn.execute('''
            CREATE TABLE IF NOT EXISTS productos (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                nombre TEXT NOT NULL,
                descripcion TEXT,
                precio REAL NOT NULL,
                fecha_caducidad TEXT,
                fecha_registro TEXT,
                codigo_barras TEXT UNIQUE NOT NULL,
                stock_actual INTEGER NOT NULL CHECK(stock_actual >= 0)
            )
        ''')
        # Crear tabla usuarios
        conn.execute('''
            CREATE TABLE IF NOT EXISTS usuarios (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                username TEXT NOT NULL UNIQUE,
                password TEXT NOT NULL,
```

Figura 27. Creación de Base de datos.

c) Desarrollo del backend

La solución del backend se desarrolló utilizando Python junto con el framework Flask, permitiendo crear una API eficiente y ligera para la gestión de las operaciones del sistema. Este enfoque garantiza una comunicación fluida entre la base de datos y el frontend, asegurando escalabilidad y rendimiento óptimo.

app.py | productos.db | instalaciones.txt

app.py > predict_demand_with_charts

```python
from flask import Flask, jsonify,Response, request, send_file,send_from_directory
from flask_cors import CORS
import sqlite3
import pandas as pd
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from datetime import datetime, timedelta
import os
from datetime import datetime
import xgboost as xgb
```

Figura 28. Librerías usadas en backend.

Método Registrar Usuarios

define una ruta POST ('/register') para registrar usuarios en la base de datos. Valida que los campos obligatorios (nombre de usuario, contraseña y correo electrónico) estén presentes, luego inserta los datos en la tabla 'usuarios', manejando posibles errores como duplicados o problemas en la inserción de datos.

```python
# Llamada función para registrar usuarios
create_table()
@app.route('/register', methods=['POST'])
def register_user():
    data = request.json
    try:
        if not all(key in data for key in ['username', 'password', 'email']):
            return jsonify({'error': 'Faltan campos obligatorios'}), 400

        with get_db_connection() as conn:
            conn.execute('''
                INSERT INTO usuarios (username, password, email)
                VALUES (?, ?, ?)
            ''', (data['username'], data['password'], data['email']))
            conn.commit()
        return jsonify({'message': 'Usuario registrado exitosamente'}), 201
    except sqlite3.IntegrityError:
        return jsonify({'error': 'El usuario o correo ya existen'}), 400
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 29. Método registrar usuarios.

Método login

Este método define una ruta POST ('/login') para autenticar a un usuario. Verifica que el nombre de usuario y la contraseña coincidan con los registrados en la base de datos; si son correctos, devuelve un mensaje de éxito, de lo contrario, devuelve un error de autenticación.

```python
#login
@app.route('/login', methods=['POST'])
def login():
    data = request.json
    username = data.get('username')
    password = data.get('password')

    try:
        with get_db_connection() as conn:
            user = conn.execute(
                'SELECT * FROM usuarios WHERE username = ? AND password = ?',
                (username, password)
            ).fetchone()
            if user:
                return jsonify({'message': 'Inicio de sesión exitoso'})
            else:
                return jsonify({'error': 'Usuario o contraseña incorrectos'}), 401
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 30. Método login.

Método Registrar Usuario

Este método maneja la ruta “PUT ('/update-user')” para actualizar los datos de un usuario en la base de datos. Valida que los campos obligatorios (id, nombre de usuario, contraseña y correo electrónico) estén presentes, luego actualiza los datos del usuario correspondiente en la tabla 'usuarios'. Si ocurre un error, devuelve un mensaje con el detalle del problema.

```python
#registrar usuario
@app.route('/update-user', methods=['PUT'])
def update_user():
    data = request.json
    try:
        if not all(key in data for key in ['id', 'username', 'password', 'email']):
            return jsonify({'error': 'Faltan campos obligatorios'}), 400

        with get_db_connection() as conn:
            conn.execute('''
                UPDATE usuarios
                SET username = ?, password = ?, email = ?
                WHERE id = ?
            ''', (data['username'], data['password'], data['email'], data['id']))
            conn.commit()
        return jsonify({'message': 'Usuario actualizado exitosamente'}), 200
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 31. Método registrar usuario.

Método Obtener Productos

Este método maneja la ruta GET ('/productos') para obtener la lista de todos los productos registrados en la base de datos. Recupera los datos de la tabla 'productos' y los devuelve en formato JSON. En caso de error, devuelve un mensaje con el detalle del problema.

```python
# Obtener todos los productos
@app.route('/productos', methods=['GET'])
def get_productos():
    try:
        with get_db_connection() as conn:
            productos = conn.execute('SELECT * FROM productos').fetchall()
        return jsonify([dict(producto) for producto in productos])
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 32. Método obtener productos.

Método Añadir nuevo Producto

Este método maneja la ruta POST ('/productos') para agregar un nuevo producto a la base de datos. Valida que los campos obligatorios (nombre, precio, código de barras y stock actual) estén presentes, permite incluir campos opcionales como descripción y fechas, y registra el producto en la tabla 'productos'. Maneja errores como duplicados en el código de barras o problemas generales en la inserción.

```python
# Añadir un nuevo producto
@app.route('/productos', methods=['POST'])
def add_producto():
    data = request.json
    try:
        if not all(key in data for key in ['nombre', 'precio', 'codigo_barras', 'stock_actual']):
            return jsonify({'error': 'Faltan campos obligatorios'}), 400

        with get_db_connection() as conn:
            conn.execute('''
                INSERT INTO productos (nombre, descripcion, precio, fecha_caducidad, fecha_registro
                VALUES (?, ?, ?, ?, ?, ?, ?)
            ''', (
                data['nombre'],
                data.get('descripcion', ''),  # Descripción opcional
                data['precio'],
                data.get('fecha_caducidad', ''),  # Fecha de caducidad opcional
                data.get('fecha_registro', ''),  # Fecha de registro opcional
                data['codigo_barras'],
                data['stock_actual']
            ))
```

Figura 33. Método añadir un nuevo producto.

Método Actualizar Productos

Este método maneja la ruta PUT ('/productos/int:id') para actualizar los datos de un producto específico en la base de datos identificado por su ID. Recibe los nuevos valores en formato JSON, permite campos opcionales como descripción y fechas, y actualiza el registro correspondiente en la tabla 'productos'. Si ocurre un error, devuelve un mensaje con los detalles del problema.



```python
# Actualizar un producto
@app.route('/productos/<int:id>', methods=['PUT'])
def update_producto(id):
    data = request.json
    try:
        with get_db_connection() as conn:
            conn.execute('''
                UPDATE productos
                SET nombre = ?, descripcion = ?, precio = ?, fecha_caducidad = ?, fecha_registro = ?
                WHERE id = ?
            ''', (
                data['nombre'],
                data.get('descripcion', ''),
                data['precio'],
                data.get('fecha_caducidad', ''),
                data.get('fecha_registro', ''),
                data['codigo_barras'],
                data['stock_actual'],
                id
            ))
            conn.commit()
        return jsonify({'message': 'Producto actualizado exitosamente'})
```

Figura 34. Método actualizar productos.

Método Eliminar Productos

Este método elimina un producto específico de la base de datos usando su ID. Si no se encuentra el producto, devuelve un mensaje de error con un código 404. Si ocurre un error durante el proceso, devuelve detalles del problema.



```python
# Eliminar un producto
@app.route('/productos/<int:id>', methods=['DELETE'])
def delete_producto(id):
    try:
        with get_db_connection() as conn:
            result = conn.execute('DELETE FROM productos WHERE id = ?', (id,))
            if result.rowcount == 0:
                return jsonify({'error': 'Producto no encontrado'}), 404
            conn.commit()
        return jsonify({'message': 'Producto eliminado exitosamente'})
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 35 .Método eliminar producto.

Método Buscar Producto por código de barras

Este método busca un producto en la base de datos utilizando su código de barras. Si el producto existe, devuelve sus datos en formato JSON; si no se encuentra, devuelve un mensaje de error con un código 404. En caso de error, retorna detalles del problema.

```python
# Buscar producto por código de barras
@app.route('/productos/buscar/<codigo_barras>', methods=['GET'])
def buscar_producto(codigo_barras):
    try:
        with get_db_connection() as conn:
            producto = conn.execute(
                'SELECT * FROM productos WHERE codigo_barras = ?',
                (codigo_barras,)
            ).fetchone()
        if producto:
            return jsonify(dict(producto))
        return jsonify({'message': 'Producto no encontrado'}), 404
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 36. Método buscar producto por código de barras.

Método Para generar Factura en pdf

Este método genera un archivo PDF con el detalle de una venta específica, incluyendo información como el ID de la venta, la fecha, el total y los productos asociados. Primero, obtiene los datos de la venta y los productos de la base de datos. Luego, crea un PDF en memoria utilizando la biblioteca **reportlab**, formatea los datos en un diseño estructurado y lo devuelve como un archivo descargable. Si la venta no se encuentra o ocurre un error, devuelve un mensaje con el detalle del problema.

```python
# Ruta para generar el PDF
@app.route('/ventas/pdf/<int:venta_id>', methods=['GET'])
def generar_pdf(venta_id):
    try:
        # Obtener datos de la venta y los productos
        with get_db_connection() as conn:
            venta = conn.execute("SELECT * FROM ventas WHERE id = ?", (venta_id,)).fetchone()
            productos = conn.execute(
                '''SELECT vp.cantidad, vp.subtotal, p.nombre, p.precio
                   FROM ventas_productos vp
                   JOIN productos p ON vp.producto_id = p.id
                   WHERE vp.venta_id = ?''', (venta_id,)
            ).fetchall()

        if not venta or not productos:
            return jsonify({'error': 'Venta no encontrada'}), 404

        # Crear el archivo PDF en memoria
        pdf_buffer = BytesIO()
        c = canvas.Canvas(pdf_buffer, pagesize=letter)
        c.drawImage("logo2.png", 50, 700, width=100, height=50)  # Logo de la empresa
        c.setFont("Helvetica-Bold", 16)
        c.drawString(200, 750, "Detalle de Venta")
```

Figura 37. Método generar PDF.

Método Actualizar Stock después de la Venta

Este método actualiza el stock de los productos después de una venta. Recibe un JSON con una lista de productos, verifica si cada producto existe en la base de datos y si tiene suficiente stock disponible. Si las condiciones se cumplen, reduce la cantidad de unidades vendidas del stock actual. Si no se encuentra el producto o el stock es insuficiente, devuelve un error específico. Finalmente, guarda los cambios en la base de datos y responde con un mensaje de éxito o un error en caso de fallo.



```python
# Actualizar el stock después de una venta
@app.route('/productos/actualizar_stock', methods=['PUT'])
def actualizar_stock():
    data = request.json
    try:
        with get_db_connection() as conn:
            for producto in data.get('productos', []):
                # Verificar que el producto existe y tiene stock suficiente
                resultado = conn.execute('''
                    SELECT stock_actual FROM productos WHERE codigo_barras = ?
                ''', (producto['codigo_barras'],)).fetchone()
                if not resultado:
                    return jsonify({'error': f"Producto con código {producto['codigo_barras']} no enc
                if resultado['stock_actual'] < producto['unidades']:
                    return jsonify({'error': f"Stock insuficiente para el producto {producto['codigo_

                # Actualizar el stock
                conn.execute('''
                    UPDATE productos
                    SET stock_actual = stock_actual - ?
                    WHERE codigo_barras = ?
                ''', (producto['unidades'], producto['codigo_barras']))
            conn.commit()
        return jsonify({'message': 'Stock actualizado correctamente'})
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 38. Método actualizar stock después de la venta.

Método de registrar una venta

El método registrar_venta permite registrar una nueva venta en la base de datos, almacenando la fecha y el total de la venta en la tabla ventas y obteniendo su ID generado. A partir de ese ID, asocia los productos vendidos en la tabla ventas_productos, vinculándolos con la venta y calculando su subtotal. Además, actualiza el stock de cada producto restando la cantidad vendida en la tabla productos. Finalmente, confirma los cambios en la base de datos con un commit y responde con un mensaje de éxito que incluye el ID de la venta registrada; en caso de error, devuelve un mensaje detallado del problema.

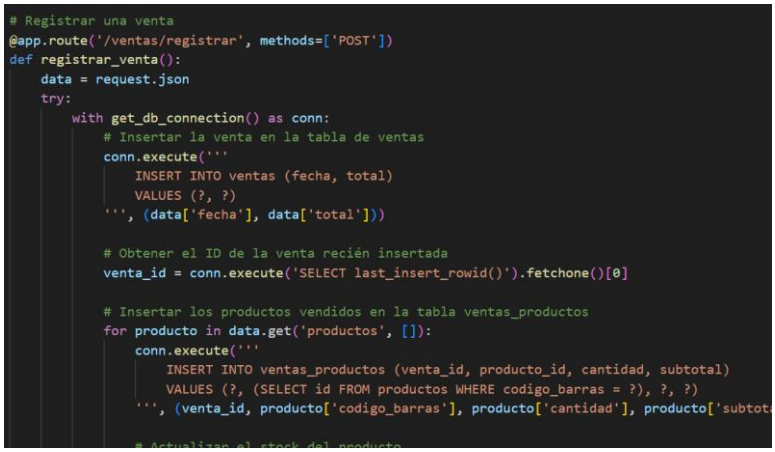


Figura 39. Método de registrar una venta.

Método para obtener detalle de Ventas Realizadas

El método get_ventas_report permite obtener un reporte paginado de las ventas registradas en la base de datos. Recibe los parámetros page y limit para determinar la página actual y la cantidad de elementos por página, y calcula el desplazamiento (offset) correspondiente. Luego, realiza una consulta a la tabla ventas para obtener los registros ordenados por fecha en orden descendente, aplicando el límite y desplazamiento especificados, además de una consulta separada para contar el total de ventas registradas. Si no hay ventas, devuelve un mensaje indicando que no se encontraron registros; de lo contrario, convierte los resultados en una lista de diccionarios con información de cada venta (venta_id, fecha, total) y calcula el número total de páginas. Finalmente, retorna la lista de ventas y el total de páginas, o un mensaje de error en caso de una excepción.

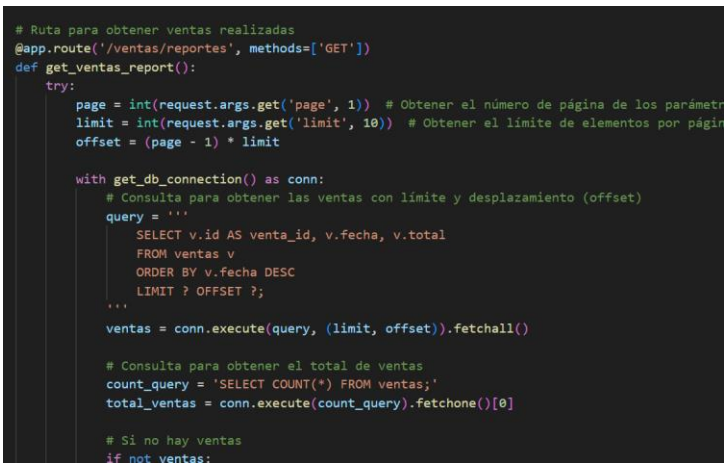


Figura 40. Método para obtener detalle de ventas realizadas.

Método para buscar productos por nombre o código de barras

El método buscar productos permite buscar productos en la base de datos por nombre o código de barras. Recibe un parámetro de consulta query, que se convierte a minúsculas para realizar una búsqueda insensible a mayúsculas y minúsculas. La consulta SQL busca en la tabla productos aquellos registros cuyo nombre o código de barras contengan el término de búsqueda. Los resultados se devuelven en formato JSON como una lista de diccionarios. Si ocurre algún error durante el proceso, se devuelve un mensaje de error con el detalle de la excepción.

```python
# Búsqueda de productos por nombre o código
@app.route('/productos/buscar', methods=['GET'])
def buscar_productos():
    query = request.args.get('query', '').lower()
    try:
        with get_db_connection() as conn:
            productos = conn.execute('''
                SELECT * FROM productos WHERE
                LOWER(nombre) LIKE ? OR
                codigo_barras LIKE ?
            ''', (f'%{query}%', f'%{query}%')).fetchall()
        return jsonify([dict(producto) for producto in productos])
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 41. Método para buscar productos por nombre o código de barras.

Método para verificar bajo stock

El método alerta_stock verifica los productos con stock inferior a 10 unidades, generando alertas para cada uno. Se conecta a la base de datos, recupera los productos y compara su stock con el umbral. Devuelve las alertas en formato JSON si hay productos con bajo stock.

```python
# Método para obtener productos próximos a caducar
@app.route('/productos/alertas_caducidad', methods=['GET'])
def alertas_caducidad():
    try:
        with get_db_connection() as conn:
            productos = conn.execute('SELECT id, nombre, fecha_caducidad FROM productos').fetchal

        today = datetime.now().date()
        threshold_days = 7  # Umbral de alerta en días
        alerts = []

        for producto in productos:
            fecha_caducidad = datetime.strptime(producto['fecha_caducidad'], '%Y-%m-%d').date()
            if today <= fecha_caducidad <= (today + timedelta(days=threshold_days)):
                alerts.append(f"Producto {producto['nombre']} caduca pronto: {fecha_caducidad}")

        return jsonify({'alertas': alerts})
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Figura 42. Método para verificar bajo stock.

Método para obtener productor cerca de caducar

Este método alertas_caducidad verifica los productos cuya fecha de caducidad está próxima, dentro de un umbral de 7 días. Conecta a la base de datos, compara las fechas de caducidad con la fecha actual y genera alertas si algún producto está próximo a caducar. Devuelve las alertas en formato JSON.

```python
#Metodo de prediccion de la demanda
@app.route('/productos/prediccion_demanda', methods=['POST'])
def predict_demand_with_charts():
    try:
        # Obtener datos del cuerpo de la solicitud
        data = request.get_json()
        codigo_barras = data.get('codigo_barras')

        if not codigo_barras:
            return jsonify({"error": "Se requiere el código de barras del producto"}), 400

        # Consultar datos históricos de ventas del producto
        with get_db_connection() as conn:
            query = '''
                SELECT v.fecha, vp.cantidad
                FROM ventas_productos vp
                JOIN ventas v ON vp.venta_id = v.id
                JOIN productos p ON vp.producto_id = p.id
                WHERE p.codigo_barras = ?
                ORDER BY v.fecha ASC
            '''
            rows = conn.execute(query, (codigo_barras,)).fetchall()

        if not rows:
            return jsonify({"error": "No se encontraron datos históricos para el producto"}), 404
```

Figura 43. Método para obtener productor cerca de caducar.

Método para crear grafico de tendencias

Este método detect_trends obtiene datos históricos de ventas de productos, agrupa las ventas por fecha y genera un gráfico de tendencias utilizando matplotlib. El gráfico se guarda en la carpeta "tendencias" con un nombre único basado en la fecha y hora. Además, se elimina cualquier gráfico antiguo para mantener la carpeta limpia. El gráfico generado es devuelto en una respuesta JSON junto con una explicación sobre el análisis de tendencias. También se proporciona una ruta para acceder al gráfico generado.



```python
@app.route('/productos/deteccion_tendencias', methods=['GET'])
def detect_trends():
    try:
        # Consultar datos históricos de ventas
        with get_db_connection() as conn:
            query = '''
                SELECT v.fecha, vp.cantidad
                FROM ventas_productos vp
                JOIN ventas v ON vp.venta_id = v.id
                ORDER BY v.fecha ASC
            '''
            rows = conn.execute(query).fetchall()

        if not rows:
            return jsonify({"error": "No se encontraron datos históricos"}), 404

        # Convertir los datos en un DataFrame de pandas
        df = pd.DataFrame(rows, columns=["fecha", "cantidad"])
        df["fecha"] = pd.to_datetime(df["fecha"])
        df.set_index("fecha", inplace=True)
        df = df.groupby("fecha").sum().sort_index()  # Agrupar por fecha y sumar cantidades

        # Generar gráfico de tendencias
        plt.figure(figsize=(14, 7))
        plt.plot(df.index, df["cantidad"], label="Histórico de ventas")
        plt.xlabel("Fecha")
```

Figura 44. Método para crear grafico de tendencias.

Método de Predicción de la demanda.

Predicción de demanda (/productos/prediccion_demanda): Este endpoint realiza predicciones sobre la demanda de productos basadas en datos históricos de ventas. Utiliza dos métodos de predicción: ARIMA para series temporales y XGBoost para regresión basada en características temporales como el día de la semana, el día del mes y el mes del año. El resultado es una predicción combinada (promediada) para los próximos 7 y 30 días. Además, genera un gráfico con las predicciones y los datos históricos de ventas.

```python
# Realizar predicciones con ARIMA para los próximos 7 días y un mes (30 días)
steps_week = 7
steps_month = 30
arima_forecast_week = arima_fit.forecast(steps=steps_week)
arima_forecast_month = arima_fit.forecast(steps=steps_month)

# Preparar datos para XGBoost
df['day_of_week'] = df.index.dayofweek
df['day_of_month'] = df.index.day
df['month'] = df.index.month
df['year'] = df.index.year

X = df[['day_of_week', 'day_of_month', 'month', 'year']]
y = df['cantidad']

# Dividir los datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# Configurar y ajustar el modelo XGBoost
xgb_model = xgb.XGBRegressor(objective='reg:squarederror')
xgb_model.fit(X_train, y_train)

# Realizar predicciones con XGBoost para los próximos 7 días y un mes (30 días)
future_dates = pd.date_range(start=df.index[-1] + pd.Timedelta(days=1), periods=steps_month, fr
```

Figura 45. Método de predicción de la demanda.

Servidor de gráficos de predicción de la demanda

Esta ruta permite servir el gráfico generado en el endpoint de predicción de demanda, lo que permite que los usuarios accedan a los gráficos como archivos estáticos.

```python
        # Crear carpeta "predicciones" si no existe
        folder_path = "predicciones"
        os.makedirs(folder_path, exist_ok=True)

        # Generar un nombre de archivo único basado en la fecha y hora actual
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        graph_path = os.path.join(folder_path, f"prediccion_{codigo_barras}_{timestamp}.png")

        # Guardar el gráfico en la carpeta "predicciones"
        plt.savefig(graph_path)
        plt.close()

        # Responder con las predicciones y la ruta del gráfico
        return jsonify({
            "prediccion_7_dias": list(zip(dates_week.strftime("%Y-%m-%d"), combined_forecast_week)),
            "prediccion_30_dias": list(zip(dates_month.strftime("%Y-%m-%d"), combined_forecast_month))
            "grafico_url": f"/predicciones/{os.path.basename(graph_path)}"  # Ajustar para ruta accesi
        })

    except Exception as e:
        return jsonify({"error": str(e)}), 500

# Habilitar la carpeta de predicciones como estática
@app.route('/predicciones/<filename>')
def servir_grafico(filename):
    return send_from_directory('predicciones', filename)
```

Figura 46. Método para generación de gráficos.

Limpiar gráficos antiguos

Este endpoint elimina todos los archivos de la carpeta de predicciones, limpiando gráficos anteriores. Es útil para mantener el almacenamiento organizado y evitar la acumulación de archivos antiguos.

```python
#eliminar fotos predicciones
@app.route('/productos/limpiar_predicciones', methods=['POST'])
def limpiar_predicciones():
    try:
        folder_path = "/predicciones"
        # Eliminar todos los archivos de la carpeta "predicciones"
        if os.path.exists(folder_path):
            for file in os.listdir(folder_path):
                os.remove(os.path.join(folder_path, file))
        return jsonify({"message": "Gráficos eliminados correctamente"}), 200
    except Exception as e:
        return jsonify({"error": str(e)}), 500
```

Figura 47. Método de limpieza de gráficos.

Método de generación de reportes en PDF

El siguiente código define un endpoint en Flask para generar un reporte en formato PDF de los productos más vendidos en diferentes periodos (día, semana, mes). La ruta /reportes/generareport_pdf consulta los datos de ventas desde la base de datos, genera gráficos de barras con los productos más vendidos en cada uno de los tres periodos, y luego crea un archivo PDF usando la librería ReportLab. El PDF incluye tablas con los productos más vendidos, gráficos correspondientes y un logotipo, todo esto estructurado con un formato adecuado. Después de generar el PDF, los gráficos temporales creados se eliminan y el PDF se envía al usuario como un archivo adjunto.

```python
@app.route('/reportes/generareport_pdf', methods=['GET'])
def generareport_pdf():
    try:
        # Conectar a la base de datos
        conn = get_db_connection()
        cursor = conn.cursor()

        # Consultar productos más vendidos
        query = '''
            SELECT p.nombre, SUM(vp.cantidad) AS total_vendido
            FROM ventas_productos vp
            JOIN productos p ON vp.producto_id = p.id
            JOIN ventas v ON vp.venta_id = v.id
            WHERE DATE(v.fecha) >= DATE('now', ?)  -- Filtra por rango
            GROUP BY p.nombre
            ORDER BY total_vendido DESC;
        '''
        (variable) data_mes: list[Any]
        data_mes = cursor.execute(query, ['-1 month']).fetchall()

        # Crear gráficos
        def crear_grafico(data, titulo, filename):
            nombres = [row[0] for row in data]
            cantidades = [row[1] for row in data]
            plt.figure(figsize=(6, 4))
            plt.bar(nombres, cantidades, color='skyblue')
            plt.title(titulo)
```

Figura 48. Método de generación de reportes en PDF.

d) Desarrollo del Frontend

La solución del frontend se desarrolló utilizando React, un framework de JavaScript que permite crear interfaces de usuario dinámicas y eficientes. Este enfoque facilita la construcción de componentes reutilizables y una experiencia interactiva, asegurando una integración fluida con el backend mediante APIs.

Se desarrollaron componentes específicos en archivos .js y .css para cada uno de los apartados necesarios del sistema. Esto permitió una organización modular del código, facilitando tanto la reutilización de elementos como la personalización de estilos para ofrecer una interfaz atractiva y funcional, para este apartado se seleccionaron fragmentos del código explicados a continuación.

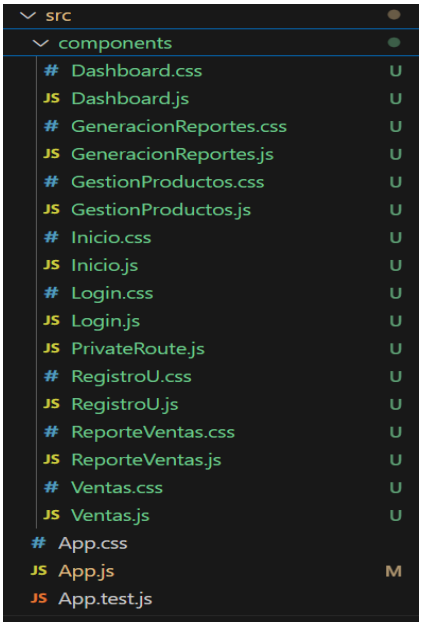


Figura 49. Componentes del frontend.

Dashboard.js

El código utilizado en esta clase corresponde al desarrollo del dashboard de la sección web, donde se crean los botones y se les asignan sus rutas correspondientes. Además, se implementa un mecanismo para garantizar la salida segura del usuario.

```
const Dashboard = () => {
    const navigate = useNavigate();

    const handleLogout = () => {
        // Limpiar el token de autenticación (si se usa)
        localStorage.removeItem('token');
        navigate('/login'); // Redirige al usuario a la página de inicio de sesión
    };

    return (
        <div className="dashboard-container">
            {/* Barra superior */}
            <header className="top-bar">
                <div className="logo-container">
                    <img
                        src={process.env.PUBLIC_URL + '/logo1.png'}
                        alt="Logo de la Panadería"
                        className="top-bar-logo"
```

Figura 50. Clase Dashboard.js.

GeneracióndeReportes.js

En esta clase se genera la parte gráfica, integrando los métodos utilizados en el algoritmo de predicción de la demanda. Además, se incluye el gráfico para la detección de tendencias y la funcionalidad para descargar el reporte de ventas.

```
import React, { useState, useEffect } from 'react';
import 'chart.js/auto'; // Asegura compatibilidad
import './GeneracionReportes.css'; // Tu archivo de estilos
import { FaFilePdf } from 'react-icons/fa';

const GeneracionReportes = () => {
  const [codigoBarras, setCodigoBarras] = useState(''); // Estado para el código de barras
  const [prediction7Days, setPrediction7Days] = useState([]);
  const [prediction30Days, setPrediction30Days] = useState([]);
  const [imageURL, setImageURL] = useState(null);
  const [trendImageURL, setTrendImageURL] = useState(null); // Estado para la URL del gráfico de t
  const [trendExplanation, setTrendExplanation] = useState(''); // Estado para la explicación del

  // Función para predecir la demanda
  const handlePrediction = async () => {
    if (!codigoBarras) {
      alert("Por favor, ingresa el código de barras del producto.");
      return;
    }
    try {
      const response = await fetch(
        "http://127.0.0.1:5000/productos/prediccion_demanda",
        {
```

Figura 51. Clase GeneracionReportes.js.

GestionProductos.js

Esta clase integra todos los métodos del backend necesarios para gestionar los productos, permitiendo agregar, editar y eliminar elementos, además de ofrecer la funcionalidad de buscar productos por código de barras.



Figura 52. Clase GestionProductos.js.

Inicio.js

Integra los métodos para generar alertas, como las de caducidad de productos y las de bajo stock.



Figura 53. Clase Inicio.js.

Login.js

Muestra gráficamente la pantalla de inicio de sesión para permitir el acceso de usuarios, además de ofrecer la opción de crear nuevos usuarios.

```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import './Login.css';

const Login = () => {
    const [username, setUsername] = useState(''); // Estado para el nombre de usuario
    const [password, setPassword] = useState(''); // Estado para la contraseña
    const [error, setError] = useState(''); // Estado para manejar mensajes de error
    const navigate = useNavigate();

    // Función para manejar el inicio de sesión
    const handleLogin = async (event) => {
        event.preventDefault();
        try {
            const response = await fetch('http://localhost:5000/login', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({ username, password }),
            });

            const data = await response.json();
```

Figura 54. Clase Login.js.

PrivateRoute.js

Permite que solo el usuario autenticado pueda navegar entre las pestañas; en caso contrario, redirige al inicio de sesión para ingresar con las credenciales correspondientes.

```
import React from 'react';
import { Navigate } from 'react-router-dom';

const PrivateRoute = ({ children }) => {
    const token = localStorage.getItem('token'); // Verifica si hay un token guardado

    // Si no hay token, redirige al login
    return token ? children : <Navigate to="/login" />;
};

export default PrivateRoute;
```

Figura 55. Clase PrivateRoute.js.

RegistroU.js

Muestra y llama el método para poder crear nuevos usuarios en el sistema.

```jsx
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom'; // Importa el hook useNavigate
import './RegistroU.css';

const RegistroU = () => {
    const [form, setForm] = useState({
        username: '',
        password: '',
        email: ''
    });

    const navigate = useNavigate(); // Inicializa el hook useNavigate

    const handleChange = (e) => {
        setForm({ ...form, [e.target.name]: e.target.value });
    };

    const handleRegister = async (e) => {
        e.preventDefault();
        try {
            const response = await fetch('http://localhost:5000/register', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify(form)
```

Figura 56. Clase RegistroU.js.

ReporteVentas.js

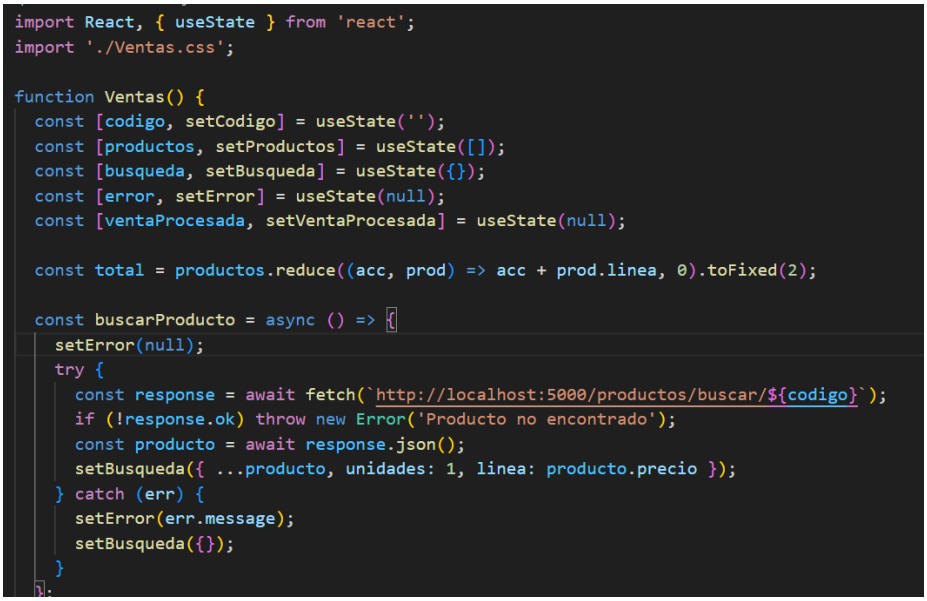
Muestra todas las ventas realizadas y ofrece la opción de descargarlas en formato XML, lo que permite tener un respaldo de los datos o utilizarlos en otros contextos, como probar nuevos modelos de machine learning.

```jsx
src > components > JS ReporteVentas.js > [@] VentasRealizadas > [@] fetchVentas
1   import React, { useState, useEffect } from 'react';
2   import './ReporteVentas.css';
3
4   const VentasRealizadas = () => {
5     const [ventas, setVentas] = useState([]);
6     const [pagina, setPagina] = useState(1);
7     const [totalPaginas, setTotalPaginas] = useState(1);
8
9     // Función para obtener ventas realizadas
10    const fetchVentas = async (pagina) => {
11      try {
12        const response = await fetch(`http://127.0.0.1:5000/ventas/reportes?page=${pagina}&limit=10`);
13        if (!response.ok) {
14          throw new Error('Error en la respuesta del servidor');
15        }
16        const data = await response.json();
17        console.log('Datos obtenidos:', data); // Añade esta línea para depuración
18        setVentas(data.ventas || []);
19        setTotalPaginas(data.totalPaginas || 1);
20      } catch (error) {
21        console.error('Error al obtener ventas:', error);
22      }
```

Figura 57. Clase ReportedeVentas.js.

Ventas.js

Este módulo incluye todos los métodos necesarios para realizar ventas de productos, con la funcionalidad de actualizar el inventario automáticamente. Además, permite descargar la venta realizada en formato PDF para entregarla a los clientes.



```
import React, { useState } from 'react';
import './Ventas.css';

function Ventas() {
  const [codigo, setCodigo] = useState('');
  const [productos, setProductos] = useState([]);
  const [busqueda, setBusqueda] = useState({});
  const [error, setError] = useState(null);
  const [ventaProcesada, setVentaProcesada] = useState(null);

  const total = productos.reduce((acc, prod) => acc + prod.linea, 0).toFixed(2);

  const buscarProducto = async () => {
    setError(null);
    try {
      const response = await fetch(`http://localhost:5000/productos/buscar/${codigo}`);
      if (!response.ok) throw new Error('Producto no encontrado');
      const producto = await response.json();
      setBusqueda({ ...producto, unidades: 1, linea: producto.precio });
    } catch (err) {
      setError(err.message);
      setBusqueda({});
    }
  };
```

Figura 58. Clase de Ventas.js.

3.2.4. Fase IV: Pruebas

En esta etapa se llevaron a cabo las pruebas de aceptación y de usabilidad, basándose en las historias de usuario establecidas durante la fase de planificación.

a) Pruebas de aceptación

Las pruebas de aceptación se realizaron con base en las historias de usuario definidas previamente, asegurando que cada funcionalidad desarrollada cumpliera con los requisitos y expectativas del usuario final. Estas pruebas consistieron en validar escenarios reales de uso, verificando la correcta interacción entre los componentes del sistema, la integración del backend y frontend, y el cumplimiento de los criterios de aceptación especificados.

Tabla 54. Prueba de aceptación de control de acceso al sistema.

Prueba de Aceptación: Control de Acceso al Sistema	
Historia de Usuario: 1	
Criterios de Aceptación	**Casos de Pruebas**
El administrador accede al sistema con credenciales válidas. El sistema muestra un mensaje de error si las credenciales son incorrectas o los campos están vacíos.	**Inicio exitoso:** Con credenciales válidas, se permite el acceso y se redirige al panel principal. **Credenciales incorrectas:** Se muestra el mensaje "Usuario o contraseña incorrectos" y el acceso es denegado. **Campos vacíos:** Se muestra "Por favor, complete todos los campos" y el acceso es denegado.
Criterios de Éxito:	
El criterio de éxito para el control de acceso se ha cumplido, al ingresar credenciales válidas, el administrador es redirigido al panel principal, mientras que, si las credenciales son incorrectas o los campos están vacíos, el sistema muestra un mensaje claro y deniega el acceso.	

Tabla 55. Prueba de aceptación de registro de nuevos usuarios.

Prueba de Aceptación: Registro de nuevos usuarios	
Historia de Usuario: 2	
Criterios de Aceptación	**Casos de Pruebas**
El sistema permite registrar un nuevo usuario con nombre, correo y contraseña válidos. Se muestra un mensaje de error si el correo ya está registrado o los datos son inválidos.	**Registro exitoso:** Con datos válidos, el usuario es creado y se muestra el mensaje "Usuario registrado con éxito". **Correo duplicado:** Si el correo ya está registrado, se muestra "El correo ingresado ya está en uso". **Datos inválidos:** Si los campos están vacíos o no cumplen los requisitos, se muestra "Datos inválidos, por favor verifique".
Criterios de Éxito:	
El criterio de éxito se cumple, al registrar un nuevo usuario con datos válidos, el sistema lo crea correctamente y muestra el mensaje "Usuario registrado con éxito", mientras que, si los datos son inválidos o el correo ya está registrado.	

Tabla 56. Prueba de aceptación de diseño del dashboard principal.

Prueba de Aceptación: Diseño del Dashboard Principal	
Historia de Usuario: 3	
Criterios de Aceptación	**Casos de Pruebas**
El administrador, tras iniciar sesión, accede al dashboard principal con el logo de la empresa en una posición destacada. El dashboard incluye botones funcionales para navegar a Inicio, Gestión de Productos, Ventas, Reportes y Reporte de Ventas. El botón de cerrar sesión funciona de manera segura. El diseño es intuitivo, responsivo y refleja la identidad corporativa de la empresa.	**Acceso al dashboard:** Tras iniciar sesión, se muestra correctamente el dashboard con los elementos solicitados (logo, botones y diseño intuitivo). **Navegación entre apartados:** Cada botón lleva al apartado correspondiente sin errores. **Cierre de sesión:** Al presionar el botón de cerrar sesión, el usuario es redirigido a la pantalla de inicio de sesión de manera segura.
Criterios de Éxito:	
Todos los elementos del dashboard funcionan correctamente y cumplen con los requisitos establecidos.	

Tabla 57. Prueba de aceptación de visualización de productos con bajo stock y próximos a caducar.

Prueba de Aceptación: Visualización de productos con bajo stock y próximos a caducar	
Historia de Usuario: 4	
Criterios de Aceptación	**Casos de Pruebas**
Al iniciar sesión, el administrador ve una lista destacada con productos con bajo stock y próximos a caducar. La sección de Bajo Stock muestra el nombre del producto y la cantidad actual. La sección de Próximos a Caducar muestra el nombre del producto, la fecha de caducidad y los días restantes.	**Bajo Stock:** En la sección de Bajo Stock, se muestra correctamente el nombre del producto y la cantidad actual disponible. **Próximos a Caducar:** En la sección de Próximos a Caducar, se muestra correctamente el nombre del producto, la fecha de caducidad y el número de días restantes antes de que caduque. **Actualización dinámica:** La lista se actualiza correctamente si hay cambios en el stock o las fechas de caducidad de los productos.
Criterios de Éxito:	
Todos los productos con bajo stock y próximos a caducar se visualizan correctamente con la información solicitada.	

Tabla 58. Prueba de aceptación de gestión de inventario.

Prueba de Aceptación: Gestión de inventario	
Historia de Usuario: 5	
Criterios de Aceptación	**Casos de Pruebas**
El administrador puede registrar, modificar, consultar y eliminar productos del inventario. Las modificaciones deben reflejarse de manera inmediata y correcta en el sistema. El sistema permite una búsqueda eficiente de productos registrados.	**Registrar producto:** El administrador puede añadir un nuevo producto al inventario, completando los campos requeridos (nombre, cantidad, precio, etc.). **Modificar producto:** El administrador puede actualizar la información de un producto existente y la actualización se refleja correctamente. **Consultar producto:** El administrador puede buscar y ver los detalles de un producto específico en el inventario. **Eliminar producto:** El administrador puede eliminar un producto del inventario.
Criterios de Éxito:	
El administrador puede realizar todas las acciones de gestión de inventario correctamente (registrar, modificar, consultar y eliminar productos).	

Tabla 59. Prueba de aceptación de gestión de ventas.

Prueba de Aceptación: Gestión de ventas	
Historia de Usuario: 6	
Criterios de Aceptación	**Casos de Pruebas**
El vendedor puede registrar ventas utilizando un escáner de códigos. El sistema calcula automáticamente los subtotales y totales de las ventas.	**Registrar venta:** El vendedor utiliza el escáner de códigos para agregar productos al carrito de ventas. **Cálculo de subtotales y totales:** Al agregar productos, el sistema calcula y muestra correctamente los subtotales y totales de la venta. **Venta exitosa:** El vendedor puede completar la venta, generar el recibo y finalizar la transacción sin errores.
Criterios de Éxito:	
El escáner de códigos funciona correctamente y los cálculos de subtotales y totales son precisos.	

Tabla 60. Prueba de aceptación de generación de nota de venta en PDF.

Prueba de Aceptación: Generación de nota de venta en PDF	
Historia de Usuario: 7	
Criterios de Aceptación	**Casos de Pruebas**
Al finalizar una venta, el sistema genera automáticamente una nota de venta con los detalles completos de la transacción. La nota de venta incluye los detalles del cliente, productos vendidos, cantidades, subtotales, impuestos y el total. El PDF está disponible para ser descargado. El diseño del PDF incluye el logo de la empresa.	**Generación de PDF:** Al finalizar una venta, el sistema genera automáticamente un archivo PDF con la información detallada de la venta. **Verificación de contenido:** El PDF contiene correctamente los detalles del cliente, productos vendidos, cantidades, subtotales, impuestos y el total. **Descarga del PDF:** El vendedor puede descargar la nota de venta en formato PDF sin problemas.
Criterios de Éxito:	
El sistema genera correctamente el PDF con toda la información necesaria y permite su descarga.	

Tabla 61. Prueba de aceptación de generación de reportes.

Prueba de Aceptación: Generación de reportes	
Historia de Usuario: 8	
Criterios de Aceptación	**Casos de Pruebas**
El administrador puede generar reportes detallados de ventas semanales y mensuales. Los reportes deben incluir datos clave como el total de ventas, productos más vendidos, y análisis de desempeño.	**Generación de reporte semanal:** El administrador selecciona la opción para generar el reporte semanal y recibe un reporte detallado con los datos correspondientes. **Generación de reporte mensual:** El administrador selecciona la opción para generar el reporte mensual y recibe un reporte detallado con los datos correspondientes. **Verificación de datos:** Los reportes deben incluir información precisa como el total de ventas, productos más vendidos y cualquier otro dato relevante para el análisis de desempeño.
Criterios de Éxito:	
Los reportes se generan correctamente con los datos requeridos, permitiendo al administrador tomar decisiones informadas.	

Tabla 62. Prueba de aceptación de Predicciones de demanda.

Prueba de Aceptación: Predicciones de demanda	
Historia de Usuario: 9	
Criterios de Aceptación	**Casos de Pruebas**
El sistema debe utilizar datos históricos para generar predicciones sobre la demanda futura de productos. Las predicciones deben mostrarse en un gráfico de tendencias fácil de interpretar.	**Generación de predicciones:** El administrador puede visualizar predicciones basadas en datos históricos de ventas, anticipando la demanda de productos. **Visualización gráfica:** Las predicciones se presentan en un gráfico de tendencias claro y comprensible, permitiendo al administrador identificar patrones y hacer ajustes de producción. **Exactitud de predicciones:** Las predicciones deben ser precisas y estar basadas en datos históricos correctamente procesados por el sistema.
Criterios de Éxito:	
El sistema muestra predicciones claras y precisas en un gráfico de tendencias, ayudando al administrador a optimizar la producción de productos.	

Tabla 63. Prueba de aceptación de visualización del historial de ventas.

Prueba de Aceptación: Visualización del historial de ventas	
Historia de Usuario: 10	
Criterios de Aceptación	**Casos de Pruebas**
El administrador puede visualizar un gráfico interactivo que muestra la tendencia de ventas a lo largo del tiempo. El gráfico debe incluir datos históricos, como la cantidad de ventas realizadas. El gráfico debe ser fácil de interpretar.	**Visualización del gráfico:** El administrador puede acceder al gráfico interactivo que muestra las ventas realizadas en un periodo determinado. **Interactividad:** El gráfico permite al administrador interactuar (por ejemplo, acercar, alejar o seleccionar rangos de fechas) para obtener detalles más específicos. **Datos históricos:** El gráfico muestra la cantidad de ventas realizadas en el tiempo.
Criterios de Éxito:	
El gráfico interactivo se presenta de manera clara y comprensible, permitiendo al administrador analizar la tendencia de ventas a lo largo del tiempo.	

b) Pruebas de Usabilidad

Se realizaron pruebas de usabilidad mediante un test con el objetivo de evaluar la facilidad de uso y la experiencia de la persona encargada de las ventas al interactuar con el sistema informático implementado para la gestión de productos y cobros. Durante estas pruebas, se observaron aspectos como la rapidez con la que los usuarios podían completar tareas comunes, la claridad de las instrucciones y la eficiencia general del sistema. Para medir el grado de satisfacción del usuario, se utilizó la escala de Likert, permitiendo obtener una visión clara sobre la efectividad y accesibilidad del sistema en su entorno de trabajo diario. Además, se recopiló retroalimentación cualitativa para identificar áreas de mejora y posibles ajustes en la interfaz o funcionalidad, con el fin de optimizar la experiencia del usuario y asegurar que el sistema se ajuste a sus necesidades y expectativas.

Escala de Likert:

- Totalmente de acuerdo.
- De acuerdo.
- Neutral.
- En desacuerdo.
- Totalmente desacuerdo.

A continuación, se presentan los resultados de la prueba realizada en la panadería y pastelería “Flor de Cebada”, en la que se evaluó el desempeño y la efectividad del sistema implementado para la gestión de productos y cobros. Durante la prueba, se analizaron diversos aspectos del funcionamiento del sistema, incluyendo la rapidez en la ejecución de tareas, la facilidad de uso por parte del personal encargado de las ventas, y la precisión en el manejo de datos. Los resultados obtenidos proporcionan una visión integral sobre el impacto del sistema en las operaciones diarias del negocio y permiten identificar posibles áreas de mejora para optimizar su rendimiento.

Tabla 64. Resultado de pruebas de usabilidad.

Resultado Pruebas de Usabilidad	
Rol: Usuario	**Código:** 1
El sistema permite realizar la gestión de productos de manera rápida y eficiente.	Totalmente de Acuerdo
La interfaz está diseñada de manera agradable y adecuada para su uso, tanto en términos de colores, botones, estructura y menú.	Totalmente de Acuerdo
El sistema permite acceder y actualizar los inventarios de productos con facilidad.	De acuerdo
El proceso de cobro es sencillo y rápido para los empleados.	Totalmente de Acuerdo
Los gráficos de las series de tiempo para la gestión de inventarios son fáciles de entender y utilizar.	De acuerdo
La descarga de reportes de ventas y productos es rápida y sencilla.	Totalmente de Acuerdo
Los módulos del sistema están organizados y estructurados de forma clara y fácil de navegar.	Totalmente de Acuerdo
En general, el manejo y uso del sistema es fácil y accesible para los empleados.	De acuerdo
El sistema proporciona retroalimentación clara en caso de errores o problemas durante su uso.	De acuerdo

Con la prueba de usabilidad realizada en panadería y pastelería “Flor de Cebada”, se concluye que el sistema informático es bien recibido por el usuario en cuanto a la facilidad de uso y diseño. La interfaz es considerada agradable y adecuada, y la organización de los módulos ayuda a evitar confusiones. La funcionalidad de gestión de inventarios y cobros es rápida y eficiente, y los reportes se generan sin complicaciones. Sin embargo, el usuario menciono que, en la parte de los gráficos de series de tiempo, la interpretación de ciertos datos podría ser más intuitiva para aquellos con menos experiencia en el análisis de datos.

3.2.5. Fase V: Implantación

En esta etapa se llevó a cabo el despliegue de la aplicación web en el entorno de producción.

Paso 1: Creación del build para el Frontend.

En la implementación del sistema web en la panadería y pastelería 'Flor de Cebada', el primer paso consistió en la creación del build de la aplicación para el frontend.

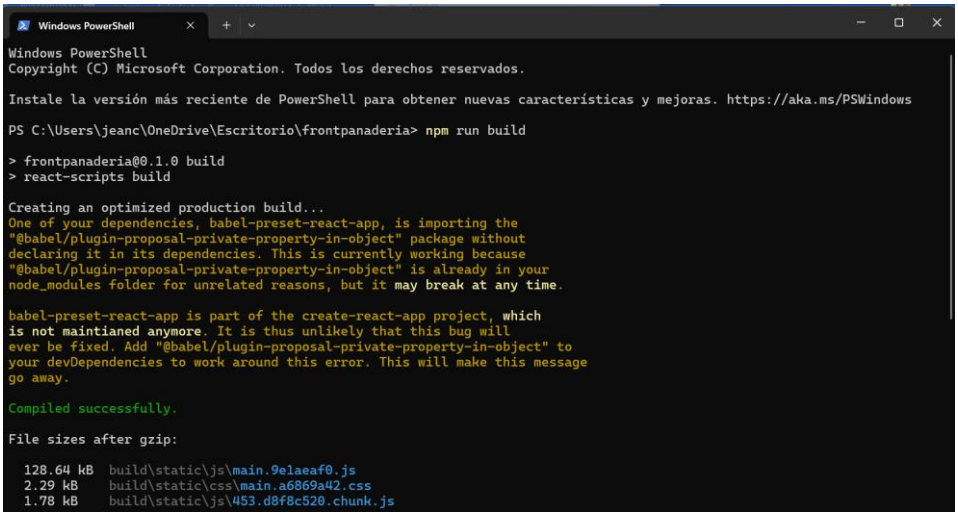


Figura 59. Creación del build para el backend.

Paso 2: Creación y configuración del servidor.

Se procedió a instalar la librería global de Node.js, lo que permitió levantar el build del frontend y configurarlo como un servidor.

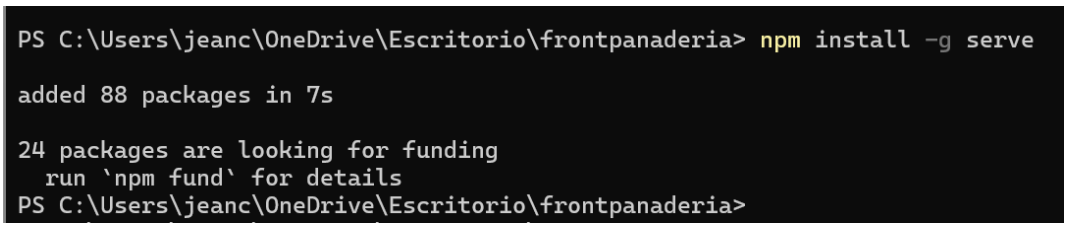


Figura 60. Instalación y configuración del servidor.

Paso 3: Automatización para poder iniciar el sistema automáticamente.

Para facilitar el despliegue del sistema, se crearon los scripts necesarios para levantar tanto el frontend como el backend de la aplicación. Estos scripts permiten iniciar de manera rápida y eficiente cada parte del sistema de forma independiente. Además, se diseñó un script adicional que permite ejecutar ambos procesos en paralelo, optimizando así el tiempo de puesta en marcha y garantizando que tanto el servidor del backend como la interfaz del frontend estén activos al mismo tiempo. Este enfoque simplifica la ejecución del sistema y mejora la eficiencia operativa del entorno de desarrollo y producción.

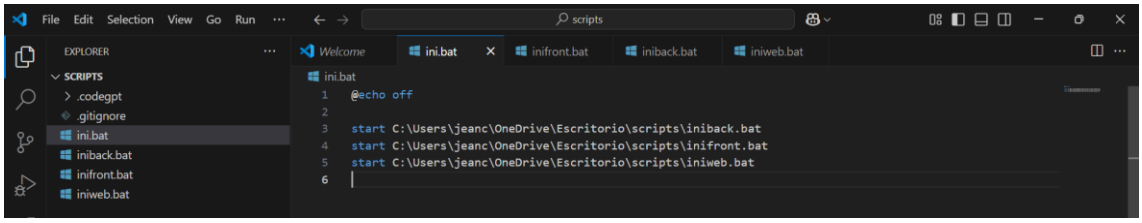


Figura 61. Script ini.bat.

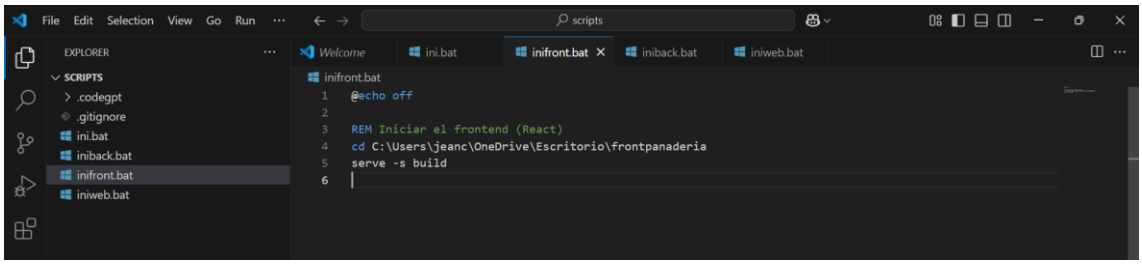


Figura 62. Script inifront.bat.

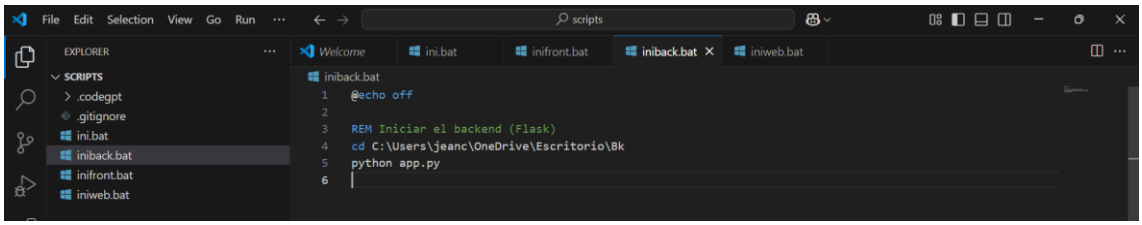


Figura 63. Script inibak.bat.

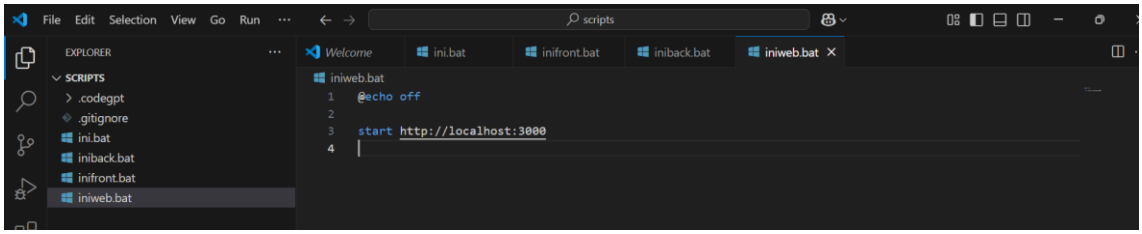


Figura 64. Script iniweb.bat.

Paso 4: Creación de una tarea programada para que el sistema encienda automáticamente cada vez que se enciende el computador.

Se configuró una tarea programada en el sistema operativo Windows para ejecutar automáticamente el archivo ini.bat cada vez que se encienda el computador. Para ello, se utilizó el Programador de Tareas de Windows, donde se creó una tarea que se desencadena al iniciar el sistema. Se especificó que la tarea ejecute el archivo ini.bat con privilegios elevados, asegurando su correcta ejecución sin necesidad de intervención manual. Esta configuración permite automatizar el proceso de ejecución del script en el arranque del sistema, mejorando la eficiencia y reduciendo la necesidad de configuraciones adicionales por parte del usuario.

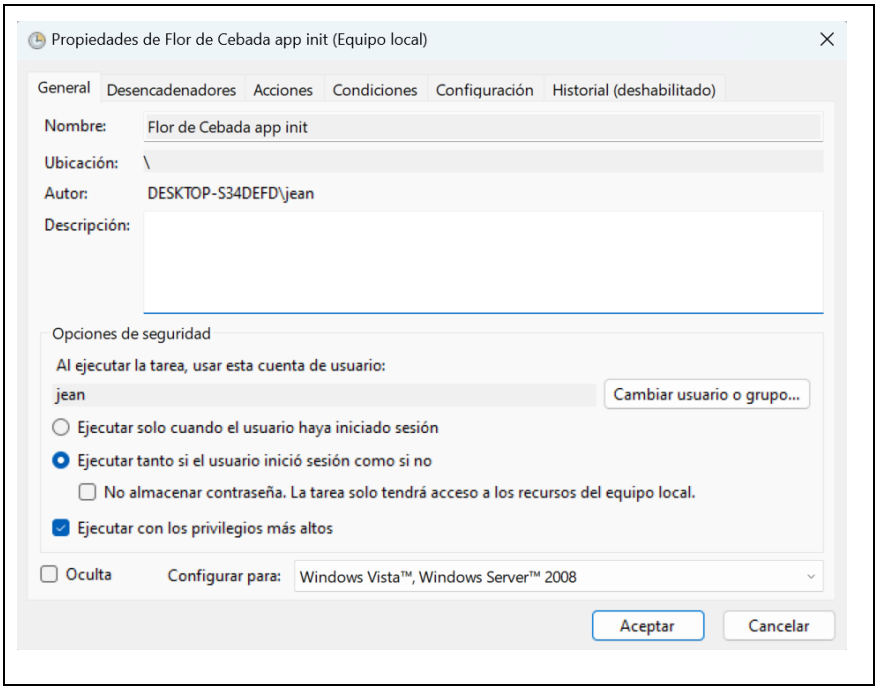


Figura 65. Configuración general de la tarea programada.

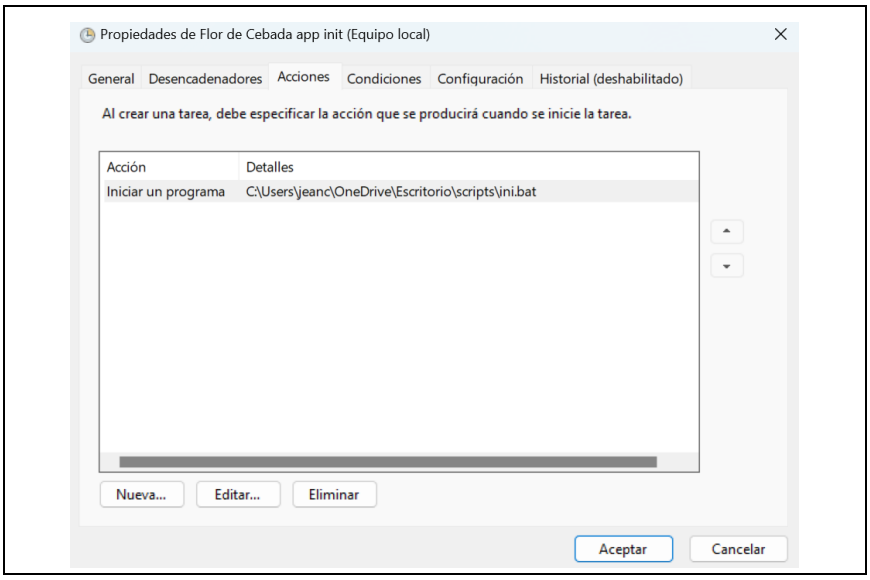


Figura 66. Acciones con la ruta para el inicio de la aplicación.

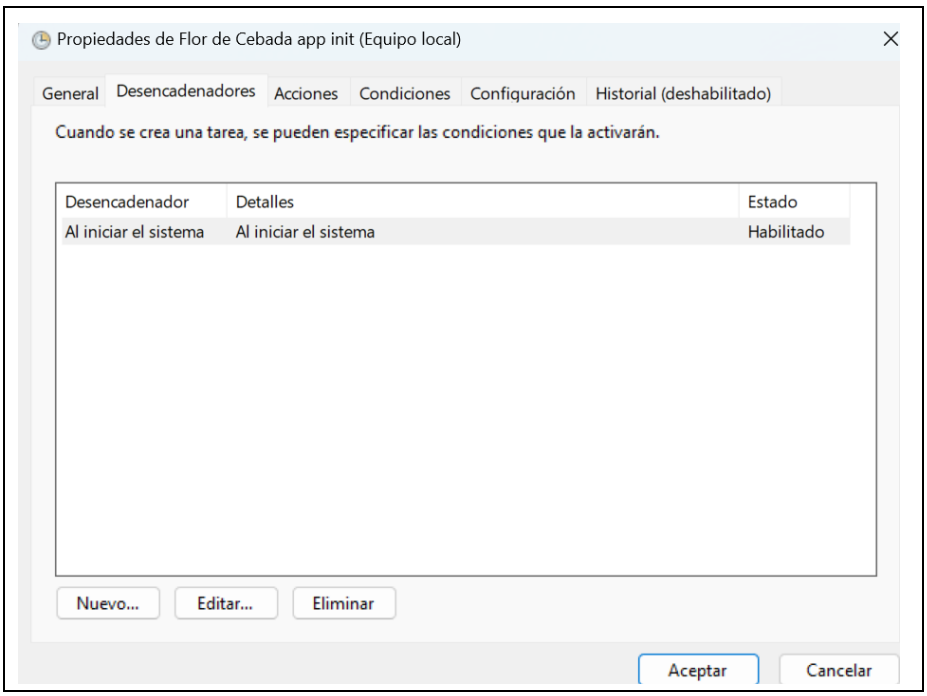


Figura 67. Desencadenador de la tarea programada.

Al reiniciar el computador, el sistema web se inicia automáticamente gracias a la tarea programada que ejecuta el archivo ini.bat, garantizando que el programa esté disponible sin intervención manual.

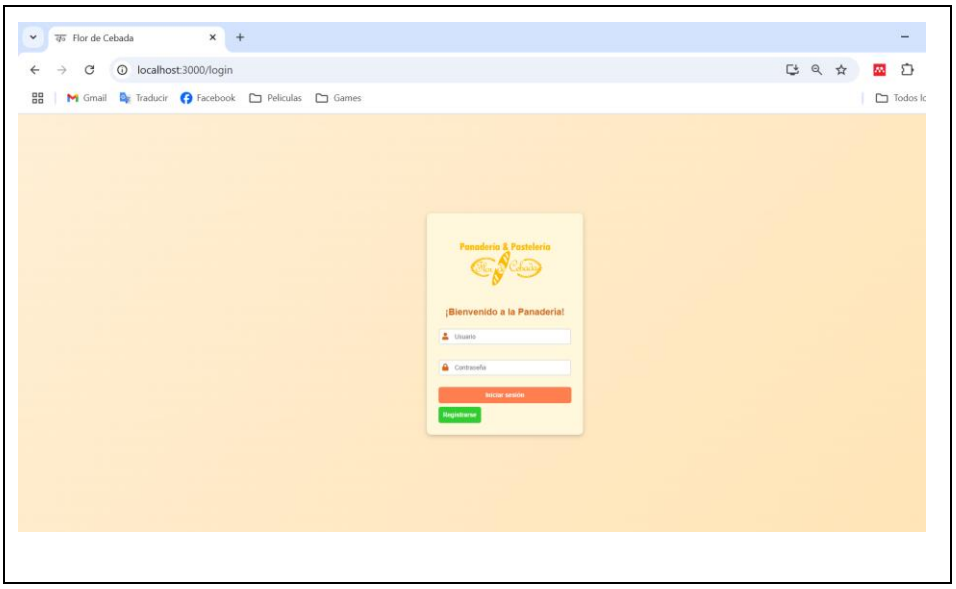


Figura 68. Inicio del sistema.

Al concluir la implementación del sistema web, se llevó a cabo un programa de capacitación dirigido al equipo de la panadería y pastelería "Flor de Cebada", con el objetivo de asegurar que el personal estuviera completamente preparado para utilizar la nueva herramienta de manera eficiente. El programa se estructuró en varias sesiones que cubrieron tanto los aspectos básicos como los más avanzados del sistema, adaptándose a las necesidades. Se brindaron instrucciones detalladas sobre cómo gestionar el inventario de productos, realizar cobros, procesar pagos, generar reportes de ventas y controlar la información financiera del negocio, entre otras funcionalidades clave del sistema.

Durante las sesiones, se fomentó la participación activa de los empleados a través de ejercicios prácticos que les permitieron interactuar directamente con el sistema en un entorno simulado, replicando situaciones reales de su día a día. Además, se incluyeron secciones de resolución de problemas comunes y manejo de incidencias, lo que permitió que los empleados adquirieran confianza y agilidad para solucionar posibles contratiempos de manera autónoma. A lo largo de la capacitación, también se enfatizó la importancia de la seguridad en el manejo de datos y la protección de la información sensible, garantizando que el equipo comprendiera cómo operar el sistema de manera segura.

Tabla 65. Capacitación a la panadería y pastelería “Flor de Cebada”.

Actividad	**Responsable**	**Dirigido a**	**Fecha**
Control de acceso al sistema	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Registro de nuevos usuarios	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Diseño del Dashboard Principal	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Visualización de productos con bajo stock y próximos a caducar	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Gestión de inventario	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Gestión de ventas	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Generación de nota de venta en PDF	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Generación de reportes	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Predicciones de demanda	Jean Carlo Freire	Personal del establecimiento.	17/12/2024
Visualización del historial de ventas	Jean Carlo Freire	Personal del establecimiento.	17/12/2024

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El análisis de los procesos internos de la panadería y pastelería “Flor de Cebada” permitió identificar áreas críticas de mejora en la gestión de productos. El sistema desarrollado automatiza estos procesos al facilitar el registro, actualización y eliminación de información relacionada con los productos, incrementando la eficiencia operativa.

- La investigación y aplicación de algoritmos de series de tiempo en Python demostró ser adecuada para predecir la demanda de productos. Esto ha facilitado una planificación más precisa, reduciendo la posibilidad de excedentes o escasez, lo que contribuye directamente a mejorar la toma de decisiones estratégicas en la producción.

- La implementación del sistema informático permitió optimizar la gestión de productos en la panadería “Flor de Cebada”, integrando predicciones de series de tiempo para mejorar la planificación. Este sistema cumplió con los objetivos planteados, brindando una herramienta eficiente y adaptable a las necesidades del negocio.

- El sistema informático desarrollado cumple con las necesidades específicas de la panadería, integrando funcionalidades clave como la predicción de demanda y la generación de reportes. Además, su diseño basado en Python, Flask, React y SQLite garantiza su accesibilidad y flexibilidad para ajustarse a futuros requerimientos.

4.2 Recomendaciones

- Establecer un cronograma para revisar los procesos internos, con el objetivo de identificar nuevas áreas de mejora y actualizar el sistema según las necesidades cambiantes de la panadería.

- Explorar y probar nuevos algoritmos o técnicas de aprendizaje automático que optimicen las predicciones de demanda, además de evaluar constantemente la calidad de estas para ajustarse a las fluctuaciones del mercado y las tendencias de consumo.

- Implementar una opción en el sistema para que los clientes realicen pedidos especiales, personalizando ingredientes y fechas de entrega. Lo que permitirá integrar esta función con el inventario y verificar la disponibilidad en tiempo real.

- Desarrollar una aplicación móvil que permita a los clientes acceder al sistema desde sus teléfonos, para realizar pedidos, consultar disponibilidad de productos en tiempo real, personalizar sus pedidos y recibir notificaciones sobre promociones o el estado de sus órdenes.

REFERENCIAS BIBLIOGRÁFICAS

[1] S. Benito Hernández et al., "Índice presentación: responsabilidad social corporativa en la industria alimentaria 4 desequilibrios socio-ambientales del sistema agroalimentario español: breve análisis y propuestas de mejora 7 alberto sanz-cobeña", Accessed: Apr. 18, 2024. [Online]. Available: www.ecosfron.org

[2] "Explora el poder de la IA: Pronóstico en series de tiempo, usando Inteligencia Artificial - UAO Portal." Accessed: Apr. 24, 2024. [Online]. Available: https://www.uao.edu.co/lo-uao-de-la-semana/explora-el-poder-de-la-ia-pronostico-en-series-de-tiempo-usando-inteligencia-artificial/

[3] Montes Paguay Sandra Viviana, "Plan de negocios para una panadería-pastelería, tipo boutique, ubicada en la parroquia la ecuatoriana, ciudad quito, ecuador.," p. 95, 2019.

[4] Y. Rios and A. Ramiro, "Facultad de ciencias empresariales carrera de comercio internacional machala 2020," 2015.

[5] H. J. Palacios Tejada and L. A. Erroa Castillo, "Ecobranding, estrategia para crear valor de marca en la industria de alimentos y bebidas en Latinoamérica.," 2022.

[6] C. DE Administración Empresas, "Escuela superior politécnica de Chimborazo facultad de administración de empresas diseño organizacional para la panadería grani's de la ciudad de Riobamba, provincia de Chimborazo," 2023.

[7] F. De Informática and Y. Electrónica, "Escuela superior politécnica de chimborazo 'sistema de cobro de agua potable de la comunidad el socorro utilizando el framework asp.net mvc'".

[8] I. G. E. G. Jaramillo Gutierrez Francisco Xavier, "Desarrollodetunsistemadecestióndeservicios para mascotas appetj," Escuela Politécnica Nacional, Quito, 2021.

[9] J. X. X. Z. Q. Z. A. S. W. CHUNLIN LI, “Fault Diagnosis Method Based on Encoding Time Series and Convolutional Neural Network,” IEEE Access, pp. 15–15, Oct. 2020.

[10] J. G. C. S. ’Julien Siebert, “A systematic review of Python packages for time series analysis,” Fraunhofer Institut for Experimental Software Engineering IESE, pp. 1–12, Jun. 2021.

[11] Sommerville Ian, Ingeniería de software. PEARSON EDUCACIÓN, 2011.

[12] E. G. ’Pacienzia, J. ’Maida, “Metodologías de desarrollo de software,” https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf.

[13] J. E. C. M. Allen Downey, “Aprenda a Pensar Como un Programador con Python,” Green Tea Press, pp. 312–312, 2002.

[14] C. Franco, A. ; Carlos, and F. Velásquez, “Cómo mejorar la eficiencia operativa utilizando el trabajo en equipo,” vol. 1, 2000, Accessed: Apr. 18, 2024. [Online]. Available: http://www.redalyc.org/articulo.oa?id=21207602

[15] Peña Sandra, Análisis de Datos. Bogotá: Fundación Universitaria del Área Andina, 2017.

[16] Avenía Delgado Carlos Arturo, Fundamentos de seguridad informática, vol. 1. Colombia: Fundación Universitaria del Área Andina, 2017.

[17] L. J. Callarisa Fiol, V. Gallart Camahort, and F. Martín Yuste, “La gestión de la cartera de productos elaboración de una guía para el análisis de la cartera de productos,” 2008.

[18] José Alberto Mauricio, Introducción al Análisis de Series Temporales, 1st ed., vol. 1. Universidad Complutense de Madrid, 2007.

[19] K. Namir, H. Labriji, and E. H. Ben Lahmar, “Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and

Combinatorial Optimization," Procedia Comput Sci, vol. 198, pp. 423–428, 2022, doi: 10.1016/j.procs.2021.12.264.

[20] U. Praveen, G. Farnaz, and G. Hatim, "Inventory management and cost reduction of supply chain processes using AI based time-series forecasting and ANN modeling," Procedia Manuf, vol. 38, pp. 256–263, 2019, doi: 10.1016/j.promfg.2020.01.034.

[21] W. Villegas-Ch, A. M. Navarro, and S. Sanchez-Viteri, "Optimization of inventory management through computer vision and machine learning technologies," Intelligent Systems with Applications, vol. 24, p. 200438, Dec. 2024, doi: 10.1016/j.iswa.2024.200438.

[22] S. K. Panda and S. N. Mohanty, "Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis," IEEE Access, vol. 11, pp. 42679–42700, 2023, doi: 10.1109/ACCESS.2023.3266275.

[23] K. P. Gaffney, M. Prammer, L. Brasfield, D. R. Hipp, D. Kennedy, and J. M. Patel, "SQLite," Proceedings of the VLDB Endowment, vol. 15, no. 12, pp. 3535–3547, Aug. 2022, doi: 10.14778/3554821.3554842.

[24] S. Raschka, J. Patterson, and C. Nolet, "Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence," Information, vol. 11, no. 4, p. 193, Apr. 2020, doi: 10.3390/info11040193.

[25] X. Zhou, W. Hu, and G.-P. Liu, "React-Native Based Mobile App for Online Experimentation," in 2020 39th Chinese Control Conference (CCC), IEEE, Jul. 2020, pp. 4400–4405. doi: 10.23919/CCC50068.2020.9189636.

[26] E. Bautista-Villegas, "Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel," Revista Amazonía Digital, vol. 1, no. 1, p. e168, Jan. 2022, doi: 10.55873/rad.v1i1.168.

ANEXOS

Anexo A manual de Usuario

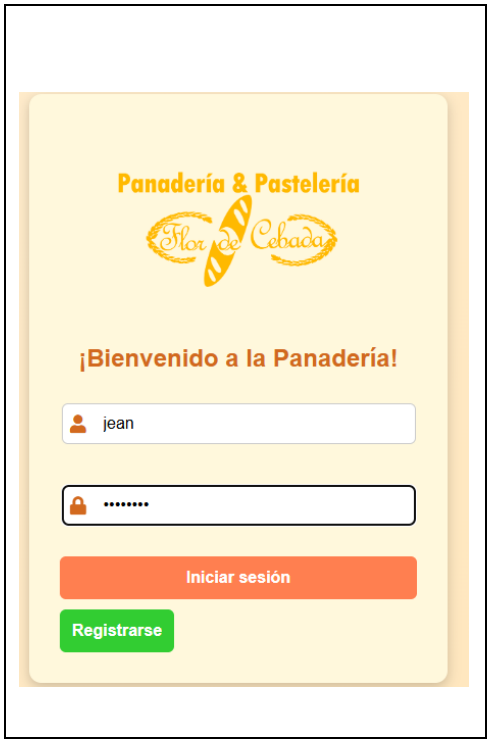
Manual de Usuario de Panadería y Pastelería Flor de cebada

Bienvenido al manual de usuario de la página web de la Panadería y Pastelería Flor de Cebada. Este manual ha sido diseñado para guiar a los encargados en el uso adecuado de nuestra plataforma de gestión de ventas e inventario, la cual facilita el proceso de control y seguimiento de los productos en nuestra panadería y pastelería.

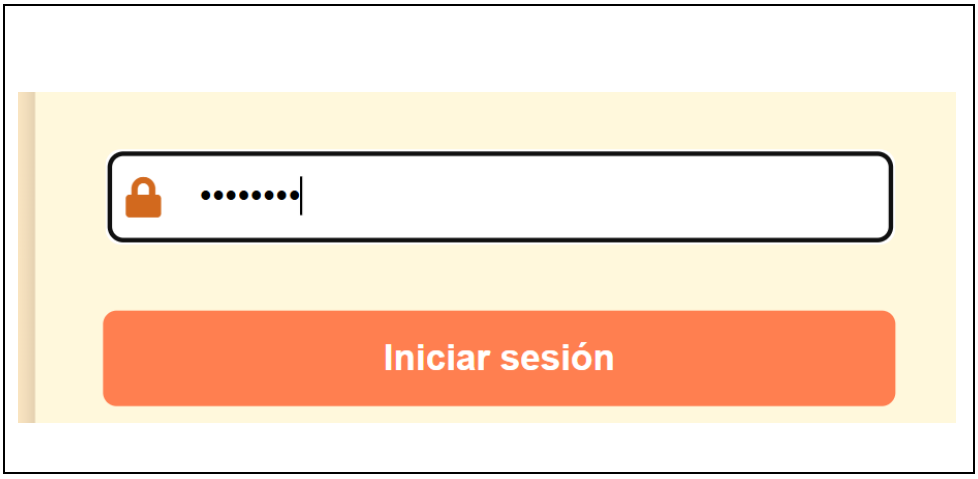
A través de esta página, los encargados podrán gestionar los pedidos de los clientes, actualizar el inventario en tiempo real y realizar un seguimiento de las ventas de manera eficiente. Además, la plataforma cuenta con herramientas para monitorear el estado de los productos, revisar el historial de ventas y hacer ajustes cuando sea necesario.

1. Ingreso al Sistema

Ingresa tu nombre de usuario y contraseña en los campos correspondientes.



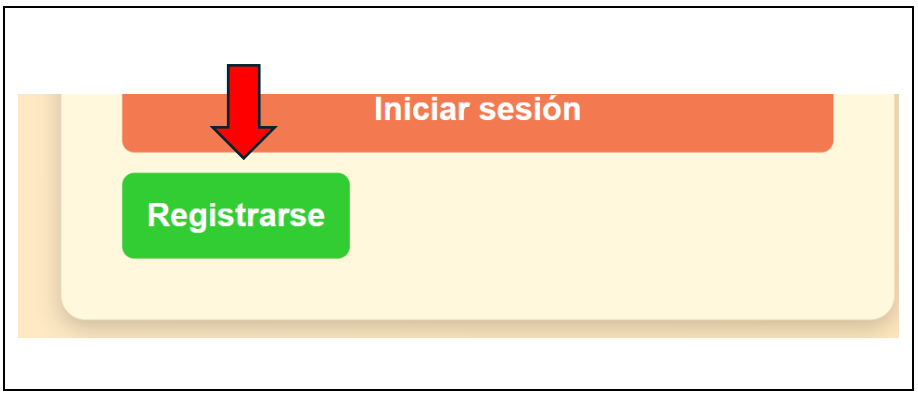
Haz clic en **"Iniciar sesión"** para acceder al sistema.



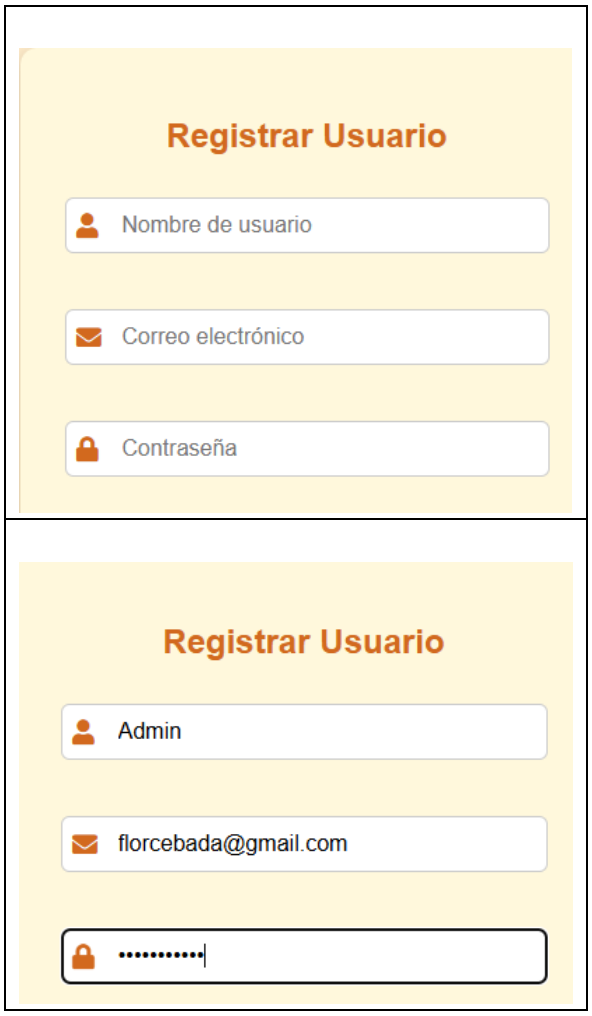
2. Registro de un Nuevo Usuario

Si no tienes un usuario para ingresar al sistema, puedes registrarte siguiendo estos pasos:

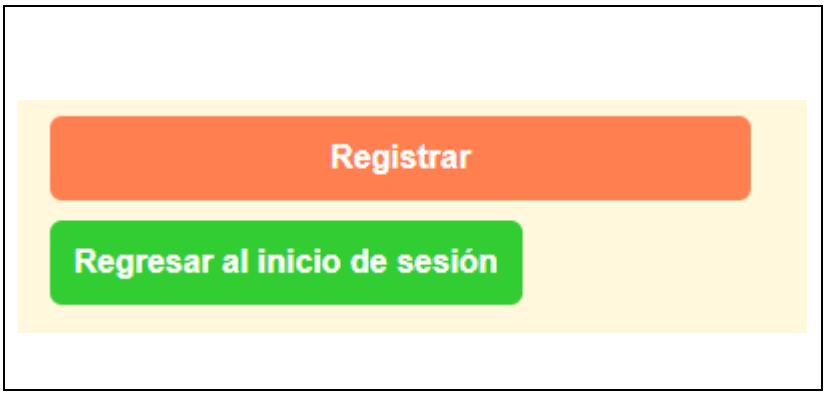
En la página de inicio de sesión, haz clic en **"Registrar nuevo usuario"**.



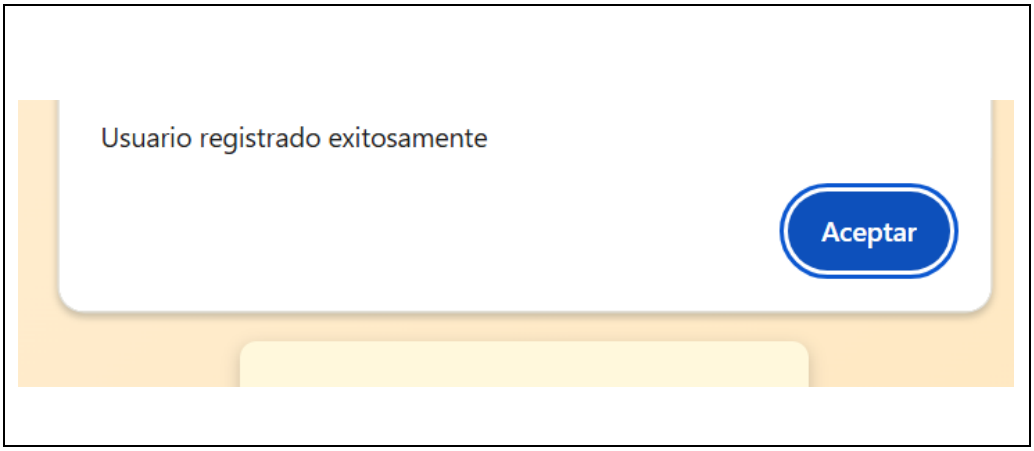
Completa el formulario con los siguientes datos.



Haz clic en **"Registrar"** para guardar los datos.



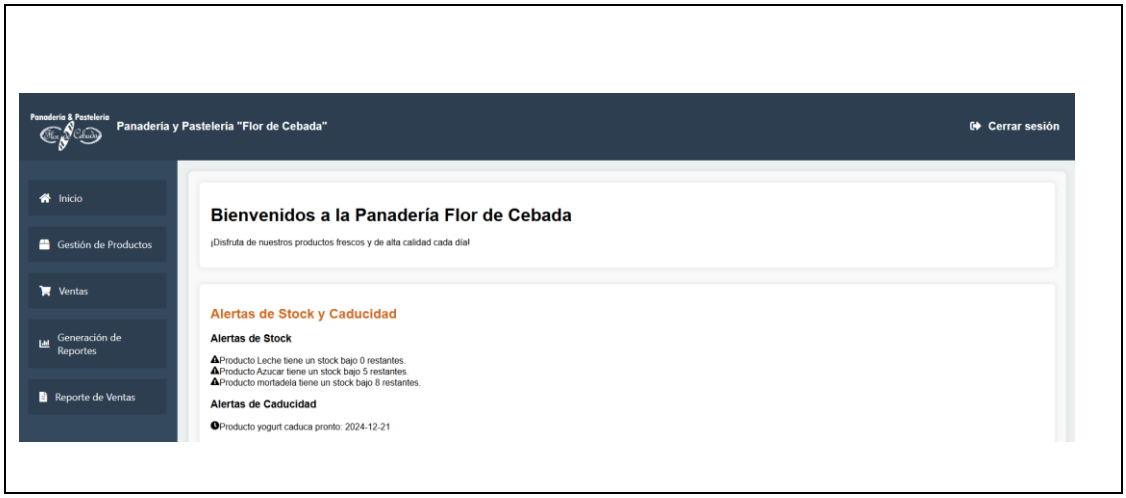
Una vez registrado, podrás acceder al sistema usando el **nombre de usuario** y **contraseña** que acabas de crear.



Nota: Si tienes problemas para registrarte, contacta al administrador para asistencia.

3. Acceso al Dashboard Principal

Una vez hayas ingresado al sistema, serás redirigido automáticamente al Dashboard Principal, donde podrás ver la información clave para gestionar el inventario y las ventas.

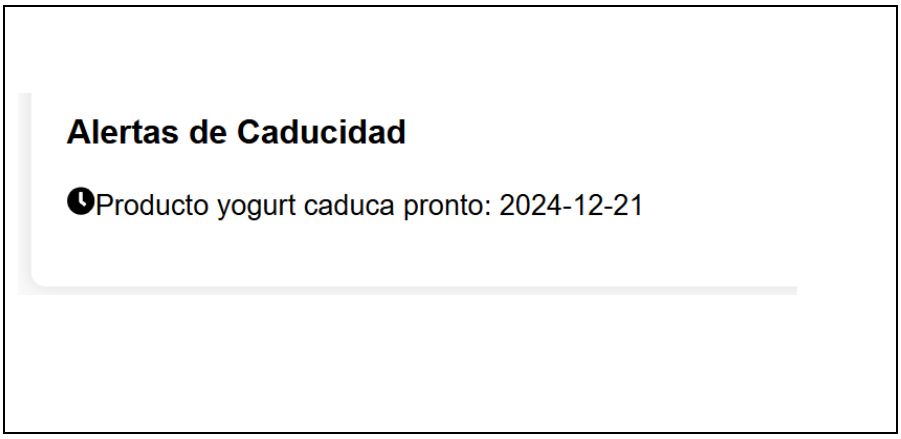


En el apartado de Inicio, encontrarás las siguientes alertas:

Alertas de Stock: Aquí se mostrarán los productos con bajo stock, para que puedas tomar las acciones necesarias antes de que se agoten.



- Alertas de Caducidad: Se indicarán los productos cercanos a su fecha de caducidad, permitiéndote gestionarlos antes de que expiren.



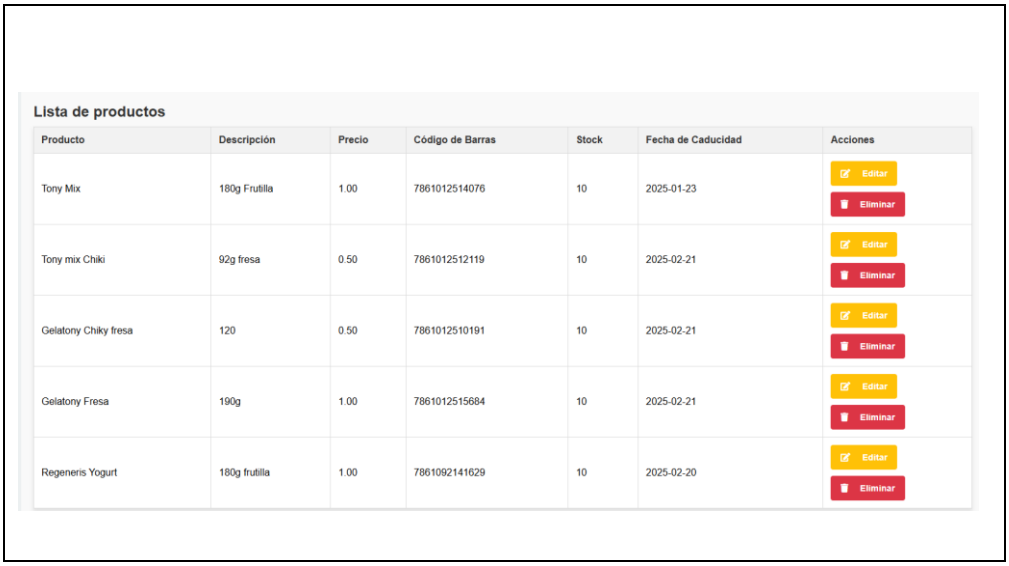
Estas alertas te permitirán estar al tanto del estado de los productos y gestionar el inventario de manera eficiente.

4. Gestión de Productos

En el apartado de Gestión de Productos, podrás ver un listado completo de los productos de la panadería, que incluye la siguiente información:

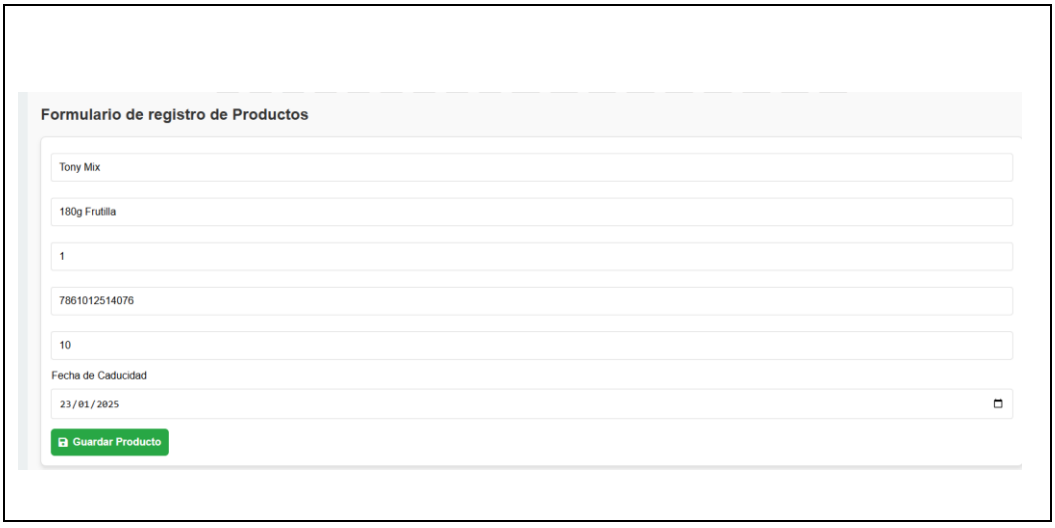
- Producto: Nombre del producto.
- Descripción: Breve descripción del producto.

- Precio: Precio del producto.
- Código de Barras: Código único del producto para su identificación.
- Stock: Cantidad disponible en inventario.
- Fecha de Caducidad: Fecha de caducidad del producto, si aplica.



Desde esta sección, puedes realizar las siguientes acciones:

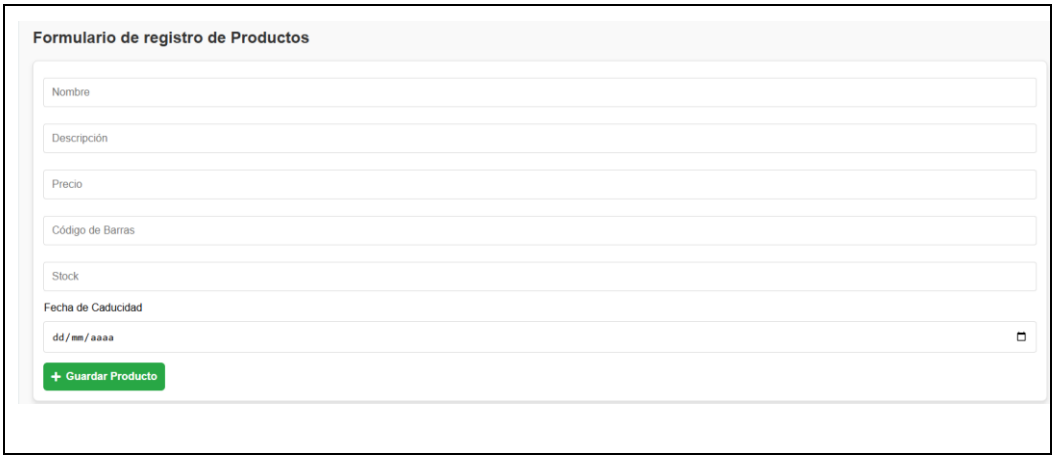
Editar: Haz clic en el icono de edición para modificar los detalles de un producto.



Eliminar: Selecciona un producto y haz clic en el icono de eliminar para borrarlo del inventario.



Agregar nuevo producto: Para agregar un producto, haz clic en el botón "Agregar Producto", completa los campos requeridos (nombre, descripción, precio, código de barras, stock y fecha de caducidad) y guarda los cambios.

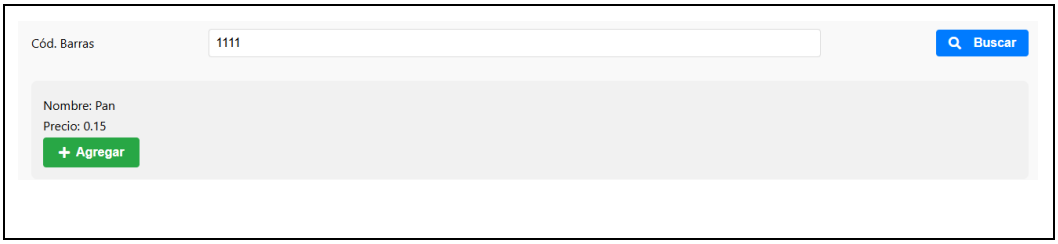


5.Ventas

En el apartado de Ventas, podrás gestionar las ventas utilizando el lector de código de barras para escanear los productos. Sigue estos pasos para realizar una venta:

- **Escanear productos:** Utiliza el lector de código de barras para escanear el código de cada producto que el cliente desee comprar. Los productos se agregarán automáticamente a la lista de venta.





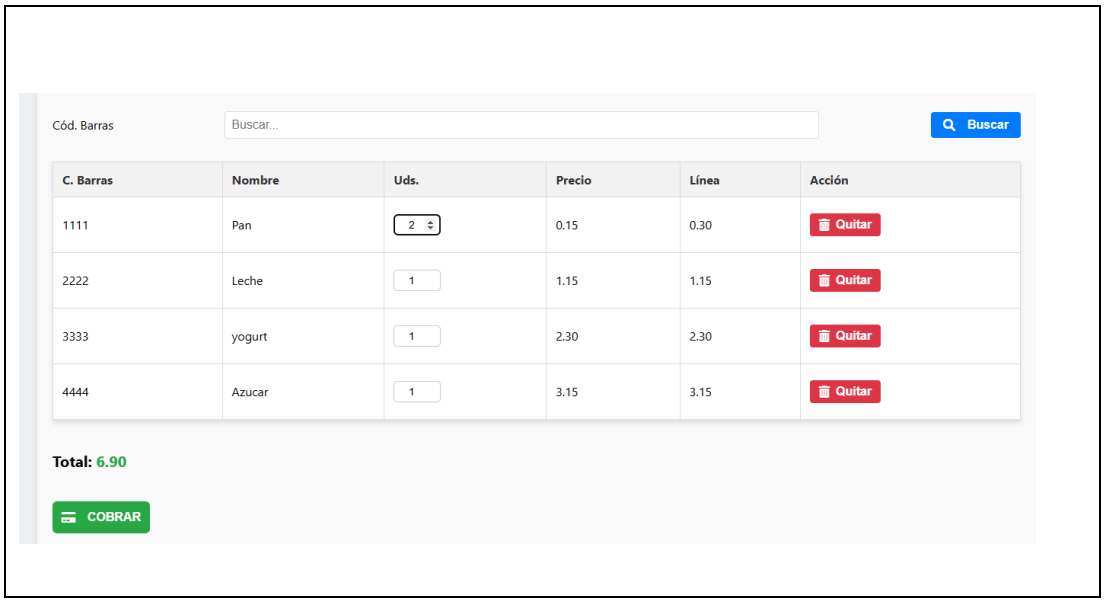
Editar unidades: Si necesitas cambiar la cantidad de unidades de un producto, puedes hacerlo directamente desde la lista de productos escaneados, editando el campo de unidades.



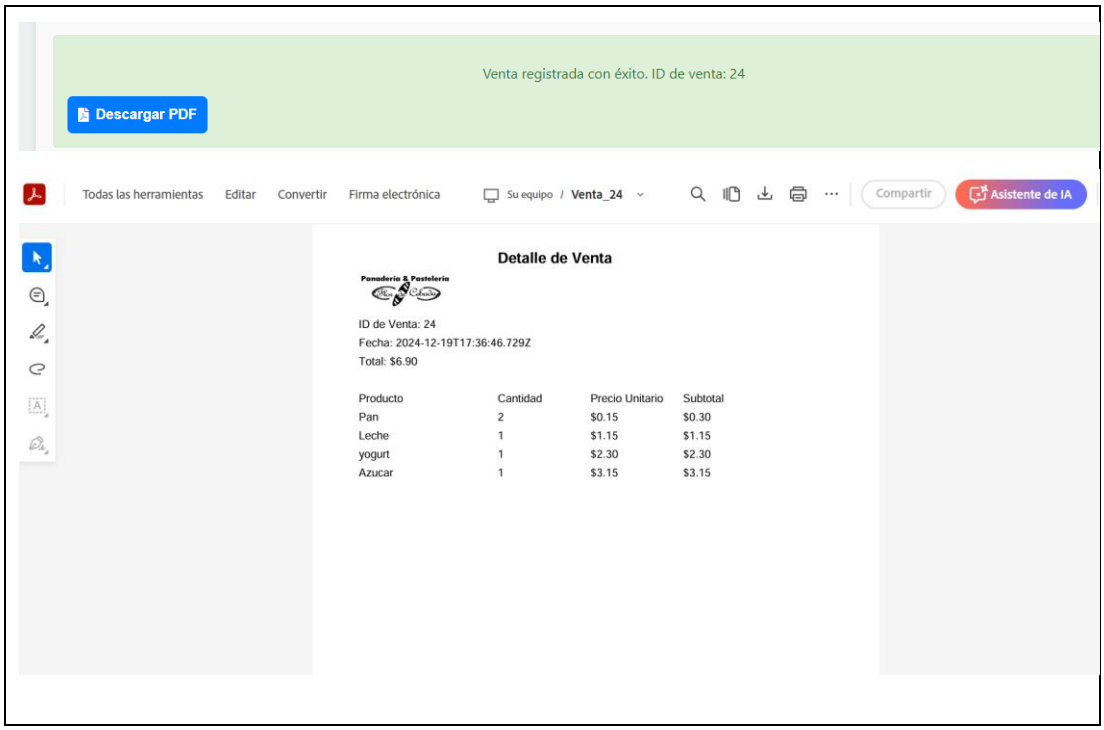
Eliminar productos: Si deseas quitar un producto de la lista, haz clic en el botón "Quitar" al lado del producto correspondiente.



Cobrar: Una vez que todos los productos estén correctos y listados, haz clic en el botón "Cobrar" para finalizar la venta.



Descargar nota de venta: Después de realizar el cobro, aparecerá un botón "Descargar Nota de Venta". Al hacer clic en él, podrás obtener el recibo de la transacción en formato PDF.



6. Generación de Predicciones

En el apartado de Generación de Predicciones, puedes obtener información valiosa sobre las ventas y tendencias de productos. A continuación, se detallan las funcionalidades disponibles:

Predicción de demanda: Puedes realizar una predicción de ventas para cualquier producto, donde se mostrará un pronóstico de la demanda para los próximos 7 días y 30 días. Esta predicción te ayudará a anticipar las cantidades de productos que necesitarás en el futuro.

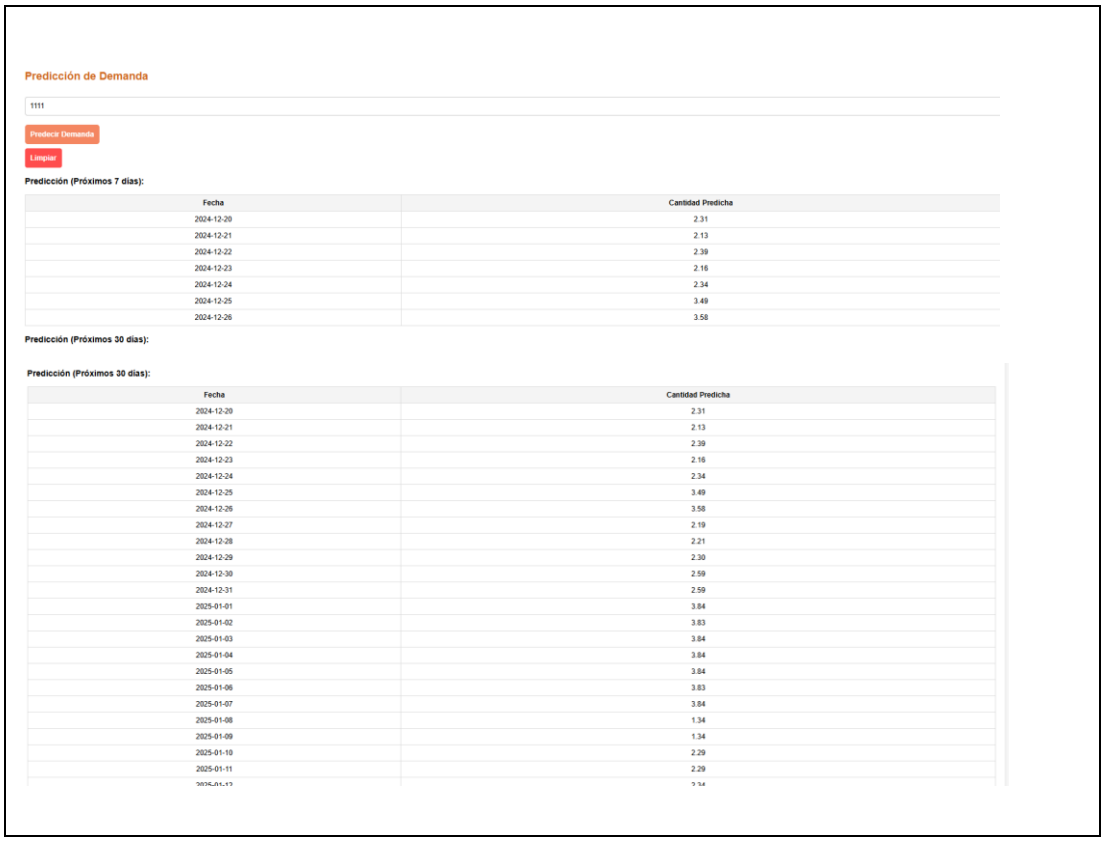


Gráfico de predicción: La predicción de ventas se presenta en un gráfico visual que representa la estimación de la demanda a lo largo del tiempo, facilitando la interpretación de los datos.

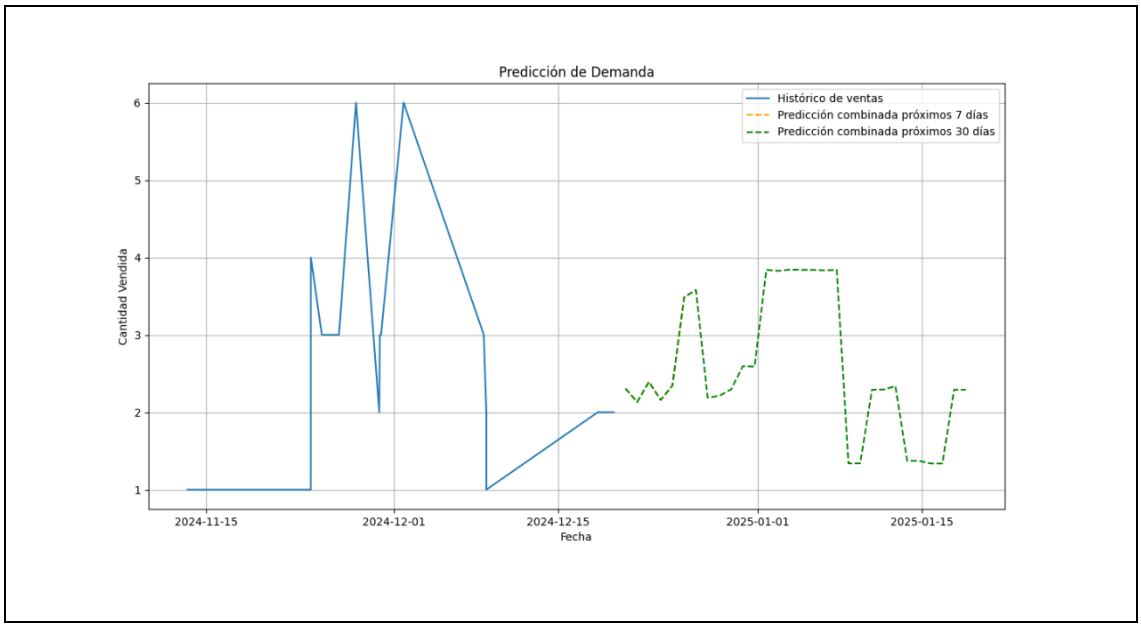
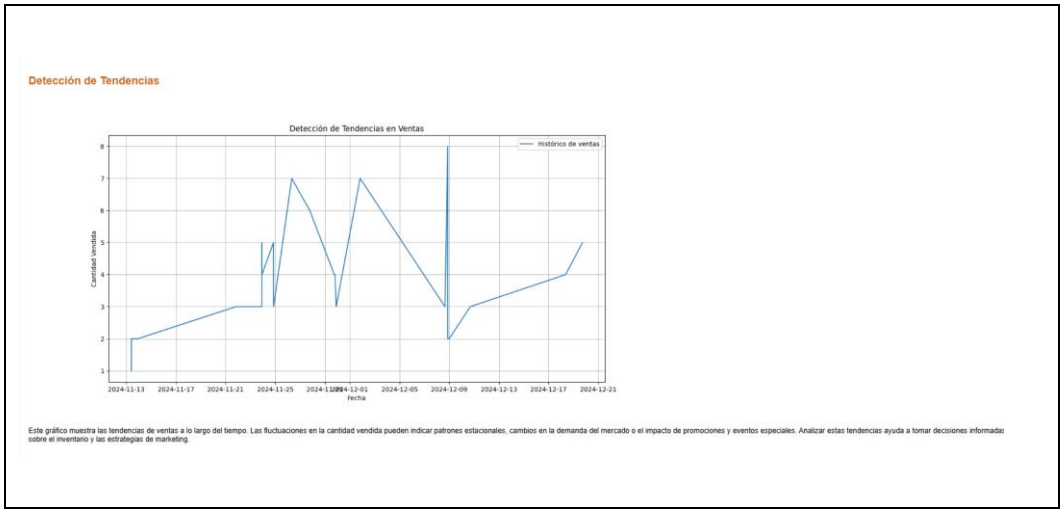


Gráfico en tiempo real de tendencias: Además, podrás visualizar un gráfico en tiempo real que muestra las tendencias de ventas, permitiéndote detectar patrones y comportamientos de compra mientras suceden.



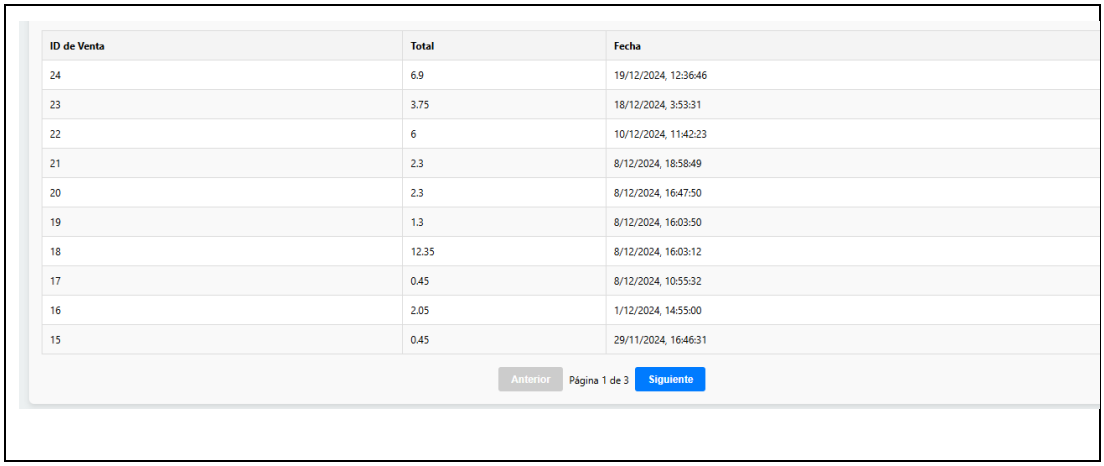
Descargar reporte de ventas: Al finalizar, puedes generar y descargar un reporte en PDF que incluye los productos más vendidos en el día, la semana y el mes, lo cual te permitirá hacer un análisis detallado de las ventas y gestionar el inventario de manera eficiente.



7. **Reporte de Ventas**

En el apartado de Reporte de Ventas, podrás visualizar todas las ventas realizadas en el sistema. Esta sección te permite acceder a un historial completo de las transacciones realizadas, con la siguiente funcionalidad:

Visualización de ventas: Se muestra un listado con todos los detalles de las ventas, incluyendo el producto, cantidad, precio y fecha de la venta.



Filtrar ventas: Puedes filtrar las ventas por diferentes criterios, como fecha, producto o cantidad, para facilitar la búsqueda de información específica.

Descargar en formato XML: Si deseas hacer uso de la información para análisis externos o integrarla con otros sistemas, puedes descargar el reporte en formato XML. Solo haz clic en el botón "Descargar XML" y obtendrás el archivo con todos los datos de las ventas realizadas.

